Technical Report

Number 815

UNIVERSITY OF CAMBRIDGE

Computer Laboratory

Colour videos with depth:
acquisition, processing and evaluation

Christian Richardt

March 2012

Some figures in this document are best viewed in colour. If you received a black-and-white copy, please consult the online version if necessary.

Summary

The human visual system lets us perceive the world around us in three dimensions by integrating evidence from *depth cues* into a coherent visual model of the world. The equivalent in computer vision and computer graphics are geometric models, which provide a wealth of information about represented objects, such as depth and surface normals. Videos do not contain this information, but only provide per-pixel colour information. In this dissertation, I hence investigate a combination of videos and geometric models: videos with per-pixel depth (also known as *RGBZ videos*). I consider the full life cycle of these videos: from their acquisition, via filtering and processing, to stereoscopic display.

Introduction & motivation

I propose two approaches to capture videos with depth. The first is a spatiotemporal stereo matching approach based on the *dual-cross-bilateral grid* – a novel real-time technique derived by accelerating a reformulation of an existing stereo matching approach. This is the basis for an extension which incorporates temporal evidence in real time, resulting in increased temporal coherence of disparity maps – particularly in the presence of image noise.

Stereo matching

The second acquisition approach is a sensor fusion system which combines data from a noisy, low-resolution time-of-flight camera and a high-resolution colour video camera into a coherent, noise-free video with depth. The system consists of a three-step pipeline that aligns the video streams, efficiently removes and fills invalid and noisy geometry, and finally uses a spatiotemporal filter to increase the spatial resolution of the depth data and strongly reduce depth measurement noise.

Time-of-flight sensor fusion

I show that these videos with depth empower a range of video processing effects that are not achievable using colour video alone. These effects critically rely on the geometric information, like a proposed video relighting technique which requires high-quality surface normals to produce plausible results. In addition, I demonstrate enhanced non-photorealistic rendering techniques and the ability to synthesise stereoscopic videos, which allows these effects to be applied stereoscopically.

Video effects

These stereoscopic renderings inspired me to study stereoscopic viewing discomfort. The result of this is a surprisingly simple computational model that predicts the visual comfort of stereoscopic images. I validated this model using a perceptual study, which showed that it correlates strongly with human comfort ratings. This makes it ideal for automatic comfort assessment, without the need for costly and lengthy perceptual studies.

Stereoscopic viewing comfort

ACKNOWLEDGEMENTS

PUBLICATIONS

This dissertation presents research that has been published in these papers:

– **Stereo coherence in watercolour rendering**
Christian Richardt, Jan Eric Kyprianidis, Neil A. Dodgson
Poster at *NPAR* and *Computational Aesthetics*, June 2010

– **Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid**
Christian Richardt, Douglas Orr, Ian Davies, Antonio Criminisi, Neil A. Dodgson
In *Proceedings of the European Conference on Computer Vision (ECCV)*, September 2010

– **Predicting stereoscopic viewing comfort using a coherence-based computational model**
Christian Richardt, Lech Świrski, Ian Davies, Neil A. Dodgson
In *Proceedings of Computational Aesthetics*, August 2011

– **Coherent spatiotemporal filtering, upsampling and rendering of RGBZ videos**
Christian Richardt, Carsten Stoll, Neil A. Dodgson, Hans-Peter Seidel, Christian Theobalt
Computer Graphics Forum (Proceedings of Eurographics), 31(2), May 2012

The following publications resulted from other work and are not presented here:

– **Flash-exposure high dynamic range imaging: virtual photography and depth-compensating flash**
Christian Richardt
Technical Report UCAM-CL-TR-712, March 2008

– **Voronoi video stylisation**
Christian Richardt and Neil A. Dodgson
In *Proceedings of Computer Graphics International (Short Papers)*, May 2009

– **Proteus – semi-automatic interactive structure-from-motion**
Malte Schwarzkopf and Christian Richardt
Poster at the *Vision, Modeling, and Visualization Workshop (VMV)*, November 2009

– **Layered photo pop-up**
Lech Świrski, Christian Richardt, Neil A. Dodgson
In *SIGGRAPH Posters*, August 2011 (*Winner of the ACM SIGGRAPH Student Research Competition*)

CONTENTS

FIGURES

Tables

Introduction **1**

The recent renaissance of stereoscopic cinema has sparked a renewed enthusiasm for stereoscopy – the art and science of fooling the human visual system into perceiving a three-dimensional image by presenting different stimuli to our two eyes. Like previous waves of stereoscopic cinema in the 1920s and 1950s, the most recent wave is driven by technological advances, specifically digital projection. This breakthrough eliminates the registration and synchronisation problems experienced with previous projection technology, and instead provides a crisp, perfectly-aligned and synchronised stereoscopic viewing experience. *Stereoscopic cinema*

However, it is not only the new technology, but also a change of mindset among stereoscopic filmmakers that drives the revival of stereoscopic cinema. Previously, filmmakers exploited the 'third dimension' by pointing props into the audience and using other gimmicks. In contrast, today's filmmakers, like James Cameron of *Avatar* fame, increasingly take a more sensible and subtle approach to stereoscopic depth in motion pictures. They view stereoscopic depth as just one technical tool of many to help them convey a film's story, for example like the choice of lighting or camera lens (Neuman, 2008; Seymour and Neuman, 2011). Instead of gimmicks, modern filmmakers aim for a more realistic, and comfortable viewing experience. *The role of depth*

Interpreting such stereoscopic imagery, and integrating the evidence from other depth cues to form a consolidated model of the visual world, comes naturally to us. Over centuries, artists have learned how to embrace this for creative expression in their paintings. In spite of all this, depth has not played a large role in image and video processing in general, and non-photorealistic rendering (NPR) in particular. As the principal aim of NPR is to create abstracted and stylised depictions of reality, not incorporating depth information appears to be a major oversight. Extending videos to encompass depth and augmenting video processing techniques to use such videos are thus the primary aims of this work. *Videos with depth*

A central theme in this dissertation is the importance of coherence – both of the temporal and stereoscopic kind: temporal coherence prevents sudden changes in videos over time, which would lead to flickering; and stereoscopic coherence avoids conflicting stimuli to be shown to both eyes as this can cause viewing discomfort. *Coherence*

1.1. Approach

Aim   The primary aim of this dissertation is to show that videos with depth (or *RGBZ videos*) provide the basis for more advanced video processing effects, for example in non-photorealistic rendering, which cannot be achieved without depth information.

Four steps   My approach to this task consists of the following four steps:

1. **Obtain depth video**
   The first step towards videos with depth is to acquire depth information in addition to a normal colour video. As there is a range of potentially suitable techniques for capturing depth information, I analyse their pros and cons before selecting stereo correspondence and time-of-flight cameras as the appropriate solutions.

2. **Filter depth video**
   Raw depth videos typically suffer from a mixture of problems such as low spatial resolution, depth quantisation artefacts, noise and flickering. Therefore, the second step concentrates on filtering the depth video to remove these artefacts and to make it temporally coherent – with the help of the existing colour video which does not suffer from many of these issues.

3. **Extend video effects**
   Once the videos with depth are of sufficient quality, they can be used to create novel video processing effects that take advantage of the depth information, such as video relighting or stereoscopic rendering from a single video with depth. Existing non-photorealistic rendering techniques are also enhanced using the depth information.

4. **Evaluate viewing comfort**
   The fourth and final step aims to evaluate the viewing comfort of stereoscopic NPR techniques. To avoid – or at least reduce – reliance on human judgments, a computational model will have to be developed which can predict viewing comfort from stereoscopic imagery alone. This step focuses on the study of stereoscopic images to prevent disruptions from time-varying imagery.

1.2. Hypotheses

By following the approach of the previous section, I aim to demonstrate that:

H1. It is possible to reconstruct dynamic scene geometry coherently at interactive frame rates.

H2. RGBZ videos facilitate a variety of advanced video processing and non-photo-realistic rendering effects.

H3. Stereoscopic viewing comfort can be predicted from stereoscopic images alone.

The common thread running through these hypotheses is my aspiration to create *computational videography* tools which provide similar creative opportunities to those afforded by computational photography for still images, and to ensure that any resulting stereoscopic renderings do not cause viewing discomfort.

1.3. Contributions

To verify the hypotheses postulated in the previous section, the publications my dissertation is based on – and by extension this dissertation itself – make multiple contributions to computer vision and graphics (Richardt et al., 2010a,b, 2011, 2012). This section summarises these contributions on a per-chapter basis.

Introduction

The main contributions of **Chapter 3** (published as Richardt et al., 2010b) are:

Stereo matching

- the reformulation of Yoon and Kweon's adaptive support weights technique as a bilateral filter (Section 3.1.2);

- the *dual-cross-bilateral (DCB) grid*, a real-time stereo correspondence technique which was the fastest at time of publication (Section 3.2.2);

- a dichromatic extension to the DCB grid which recovers precision (Section 3.2.3);

- a spatiotemporal extension to the DCB grid that incorporates temporal evidence in real time (Section 3.4); and

- five synthetic stereo videos with ground truth disparity maps that enable quantitative evaluation of video-based stereo matching techniques (Section 3.4.2).

Chapter 4's contributions (published as Richardt et al., 2012) are:

Time-of-flight

- a prototype of a computational RGBZ video camera which augments a regular video camera with a synchronised time-of-flight camera (Section 4.1);

- an efficient geometry invalidation and multi-resolution fill-in procedure for handling stereo half-occlusions and depth camera artefacts (Section 4.2); and

- a spatiotemporal filtering approach tailored to depth cameras to increase the resolution of depth data and strongly reduce noise (Section 4.3).

The specific contributions of **Chapter 5** (published as Richardt et al., 2012) are:

Video effects

- a simple relighting technique for RGBZ videos (Section 5.2);

- depth-enhanced non-photorealistic rendering techniques that extend video abstraction and stroke-based rendering to use RGBZ videos (Section 5.3); and

- a rendering technique for RGBZ videos that creates stereoscopic RGBZ videos, with demonstration of stereoscopic non-photorealistic rendering (Section 5.4).

The contributions of **Chapter 6** (published as Richardt et al., 2010a, 2011) are:

Stereo comfort

- the first computational model for predicting the visual comfort of stereoscopic images which is suited for automatic comfort assessment, without costly and lengthy perceptual studies (Section 6.3);

- a taxonomy of stereo coherence issues which affect the stereoscopic viewing comfort of human observers (Section 6.5); and

- computational tools to detect and localise such issues (Section 6.6).

1.4. Structure

This dissertation is structured as follows:

Introduction   **Chapter 1** provides an introduction to this dissertation by describing the goal of my work, outlining the approach taken to prove my hypotheses and summarising my contributions to research.

Background   **Chapter 2** reviews technical background material that is the basis for the work in this dissertation. The chapter furthermore provides a historic perspective of some of the topics, and introduces common notation.

Stereo matching   **Chapter 3** reformulates and accelerates a stereo correspondence technique so that a spatiotemporal extension can incorporate temporal evidence in real time to produce temporally coherent disparity maps.

Time-of-flight   **Chapter 4** augments a video camera with a time-of-flight sensor, and develops a data filtering approach that removes typical artefacts in the depth data and applies an efficient spatiotemporal denoising and upsampling scheme.

Video effects   **Chapter 5** demonstrates a selection of video processing effects that critically rely on depth information, and are thus unobtainable from a colour video alone, which illustrates the benefits of videos with high-quality depth information.

Stereo comfort   **Chapter 6** studies the effects of non-photorealistic rendering techniques on the viewing comfort of stereoscopic imagery by creating and evaluating a computational model which predicts stereoscopic viewing comfort.

Conclusion   **Chapter 7** concludes the dissertation by summarising the contributions of my work, revisiting the hypotheses of this chapter, and proposing possible avenues for future research.

Anaglyph glasses   In this dissertation, I show stereoscopic images as red-cyan anaglyph images (see Section 2.2.3). These images are indicated by following small red-cyan glasses: 👓. This dissertation should contain a set of paper glasses suitable for viewing these images. For the best result, please view the anaglyph images on a digital display.

Technical background   2

> " A coupla months in the laboratory can save a coupla hours in the library.
>
> — *Westheimer's discovery* "

The work in this dissertation touches on a range of topics from different disciplines. This chapter provides the technical background for the four subsequent chapters by introducing notation and reviewing key techniques.

I discuss the following topics:

2.1. **Non-photorealistic rendering**
This section aims to give a broad overview of non-photorealistic rendering techniques for abstraction and stylisation of images, video and geometry, with a focus on stereoscopic techniques.

2.2. **Human depth perception**
This section describes how the human visual system combines several depth cues into a visual model of the world around us, and how stereopsis and stereoscopy were first discovered in the Victorian era.

2.3. **Capturing dynamic geometry**
This section discusses different approaches for recovering the shape of dynamic geometry and establishes the optimal approach for RGBZ videos.

2.4. **Stereo correspondence**
This section introduces the field of stereo computer vision, and describes the standard components of correspondence techniques and how they are evaluated and ranked objectively.

2.5. **Bilateral filtering**
This section gives a brief introduction to the bilateral filter – the most common edge-preserving filter – and some of its applications.

2.1. Non-photorealistic rendering

Introduction Non-photorealistic rendering (NPR) is an area of computer graphics that studies a wide variety of artistic styles for expression and abstraction (Agrawal, 2009). It originally emerged in the early 1990s when the computer graphics community started to challenge the predominant paradigm of photorealism which had underpinned computer graphics throughout most of its history.

Historical parallels This modern departure from the pursuit of (photo-)realism parallels the emergence of impressionism in late 19th century France. Back then, the steady advancement and proliferation of photography threatened the existence of painters, because photography produced lifelike images more efficiently and reliably than painters were able to. However, the impressionists soon focused on the one thing they could inevitably do better: to offer a subjective alternative to the photograph. This was a stark break with previous art movements and – in a sense – the first conscious effort towards non-photorealistic rendering.

Objective In computer graphics, the endeavour to generate photorealistic imagery resulted in pioneering techniques such as ray tracing and radiosity which create photorealistic images from computer models. By contrast, non-photorealistic rendering aims to create abstracted and stylised depictions of computer models or the real world. For this purpose, non-photorealistic rendering combines techniques from computer vision, image processing and computer graphics.

Abstraction & stylisation The principal motivations of abstraction and stylisation differ significantly, which is nicely illustrated by the two paintings in Figure 2.1. Abstraction is concerned with removing superfluous detail and communicating the essence of a scene or object. So while Cézanne omitted unimportant detail, he succeeded in capturing the rough shape of objects and the variation in lighting. On the other hand, stylisation is all about creating aesthetic imagery and exploring novel artistic techniques. A great example is van Gogh's *magnum opus*, in which he creates a sense of energy purely from the placement of brush strokes in dynamic 'waves'.

abstraction   stylisation

Paul Cézanne
Hamlet at Payannet, near Gardanne, 1885–1886
The White House, Washington, D.C.

Vincent van Gogh
The Starry Night, 1889
Museum of Modern Art, New York City

Figure 2.1: Examples of abstraction and stylisation from post-impressionism.

The origin of the term 'non-photorealistic rendering' is not entirely clear, but it most likely goes back to a paper by Winkenbach and Salesin (1994) on pen-and-ink illustration. In any case, it is not immediately obvious why this field has been named after what it is not, and Gooch et al. (2010) perhaps best echo my sentiment: *Origin of the term*

> " There has been considerable discussion on the proper naming of the field. The term NPR (especially in its abbreviated state) seems overly general, including all rendering which does not have a photorealistic purpose. Some other names have been put forward, such as "Stylized Rendering" or "Expressive Rendering", but we feel that those terms are not inclusive enough to cover all of the material that currently fits under the purview of NPR.
>
> For better or for worse, the field in which we work is now known widely as non-photorealistic rendering, and while self-examination can indeed be a worthy pursuit, the amount of energy devoted towards the semantics of a new name for an already established field could perhaps be better spent pushing the discipline in directions that will lead to exciting new discoveries.
>
> — *Gooch, Long, Ji, Estey, and Gooch (2010)* "

As an evolving field, non-photorealistic rendering is a moving target, and every once in a while, a group of researchers surveys the state of the art to provide an overview of techniques and to inform future work. Gooch and Gooch's book (2001) provides a broad overview of many techniques, but more often than not refers to the original papers for further details. A more detailed introductory text is the book by Strothotte and Schlechtweg (2002) which covers many non-photorealistic rendering techniques with well-structured explanations, meaningful figures and pseudo code. Furthermore, no less than three SIGGRAPH courses have shed light on the field over the years (Green et al., 1999; Sousa et al., 2003; McGuire et al., 2010). Most recently, Collomosse and Kyprianidis (2011) provided a tutorial with the focus on artistic stylisation of images and video. *State of the art*

In addition to the categorisation into abstraction or stylisation, non-photorealistic rendering techniques also vary in the data they operate on: *Classification by underlying data*

- **Geometric models** are the most comprehensive data available as they provide access to many useful surface properties such as normal vectors and curvature.

- **Images** are more limited in that they only provide a per-pixel colour and no additional information about what is being depicted in the image.

- **Videos** are essentially moving images, with a new frame every few milliseconds. The main challenge is to prevent flickering and to ensure temporal coherence.

The remainder of this section summarises a small selection of key papers in non-photorealistic rendering in the areas of abstraction (Section 2.1.1) and stylisation (Section 2.1.2), and lastly with a focus on techniques operating in stereoscopic 3D (Section 2.1.3), which is relevant to the stereoscopic rendering and viewing comfort work in this dissertation (Section 5.4 and Chapter 6). *Structure of this section*

2.1.1. Abstraction

Introduction The primary intention of abstraction is to visually distill the essence of a scene's appearance by removing perceptually irrelevant or unimportant details while at the same time emphasising salient image features such as strong edges and contours. Figure 2.2 shows an outstanding illustration of this process of visual abstraction. This section discusses three classes of visual abstraction, ranging from concrete to abstract: image-based abstraction, cartoon-like abstraction, and line drawings.

concrete ——————— visual abstraction ——————→ abstract
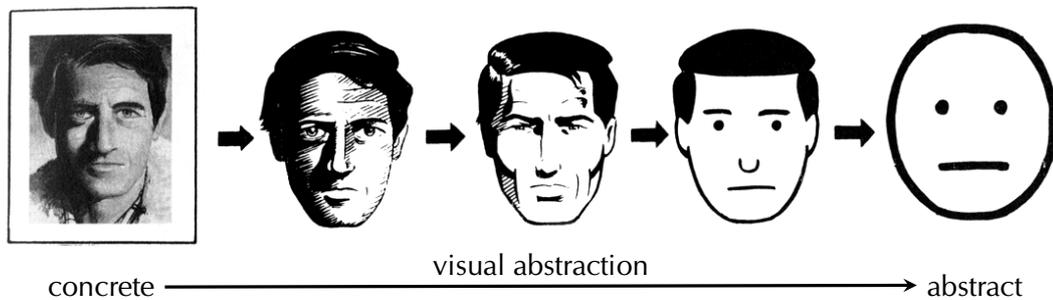
Figure 2.2: Successive reduction in visual detail from the photograph of a man to a generic smiley. Adapted from Winnemöller (2011). Illustration ©1993 Scott McCloud.

Image-based abstraction

Improved immersion in augmented reality Fischer et al. (2005) pioneered an abstracted look for augmented reality with the aim of achieving a similar look for the real and the virtual by abstracting the video and overlaying toon-shaded virtual objects. To achieve real-time frame rates, they bilaterally filter (Section 2.5) a downsampled version of the video, and upsample it. On top, they draw thick, dilated Canny edges (1986). While the abstracted video looks blurry and rough (Figure 2.3), the style set the stage for subsequent work.

Real-time video abstraction Winnemöller et al. (2006) introduced several technical and artistic improvements for video abstraction. They first iteratively apply a separated bilateral filter to preserve strong contrasts and smooth low contrast regions. They follow this with a soft luminance quantisation method that creates a stylised, cartoon-like look. Finally, difference-of-Gaussian (DoG) edges are overlaid to further increase contrast in high contrast regions. The key to temporal coherence and real-time performance is per-frame processing and making each step sufficiently temporally coherent.

Scale space analysis Orzan et al. (2007) take a different approach: they identify image edges at different resolutions and link them up into a scale space hierarchy. They then use gradient domain techniques to reconstruct an abstracted image from the truncated edge hierarchy. The results preserve prominent edges and fill the image with smooth gradients, but their implementation is very slow (10 min for one 800×600 image).

Flow-based filtering The technique by Winnemöller et al. (2006) has inspired several follow-up papers which use a variety of flow-based filtering approaches for increased visual effect. Kyprianidis and Döllner (2008) use orientation-aligned separated bilateral filtering and flow-based DoG edges to simultaneously improve computational efficiency and visual quality of the abstracted videos. Kang and Lee (2008) use a different
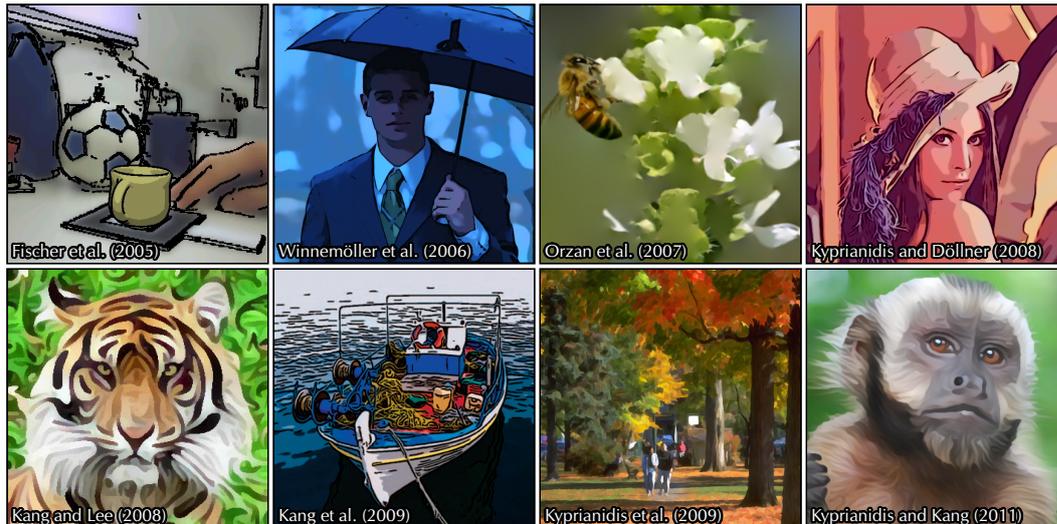
Figure 2.3: Examples of image-based abstraction techniques. © The respective copyright owners.

flow formulation and also apply shock filtering (Osher and Rudin, 1990) to prevent object boundaries from shrinking and to strengthen strong contrast edges. Kang et al. (2009) use a framework similar to Kyprianidis and Döllner, but with yet another flow formulation.

Most of the described techniques rely on the bilateral filter for visual abstraction. *Alternative filters* Other filters have been explored, such as the Kuwahara filter, which was extended by Kyprianidis et al. (2009) to adapt to the shape and orientation of local features. The result looks painterly and has the benefit of being temporally coherent. More recently, Kyprianidis and Kang (2011) have explored line integral convolution to smooth directional features, with directional shock filtering for a sharpening effect. This technique is also temporally coherent when applied on a per-frame basis.

Cartoon-like abstraction

This class of techniques shares the visual style of cartoons which is characterised by *Introduction* large areas of uniform colour, often with stylised highlights or shading. Originally, cartoons were painted on transparent 'cels' (for celluloid) and several cels were literally overlaid to create a composite scene. A number of techniques have been proposed to create cartoon-like abstraction results from 3D models, images and videos, and the key techniques are briefly described here and shown in Figure 2.4.

The process of rendering cartoon-like imagery from geometric models is generally *Cel shading* referred to as 'cel shading' or also 'toon shading'. Decaudin (1996) described the *3D models* first such rendering system, in which the diffuse shading is thresholded to create the large areas of uniform colour typical for cartoons. In addition, silhouette and contour outlines are overlaid to create the distinctive look. Gooch et al. (1998) remap the diffuse shading onto a colour ramp from cold to warm colours for automatic scientific illustration. Barla et al.'s X-Toon system (2006) introduced more flexible 2D toon textures, which are indexed by diffuse shading ($\mathbf{n}\cdot\mathbf{l}$) and tone detail.
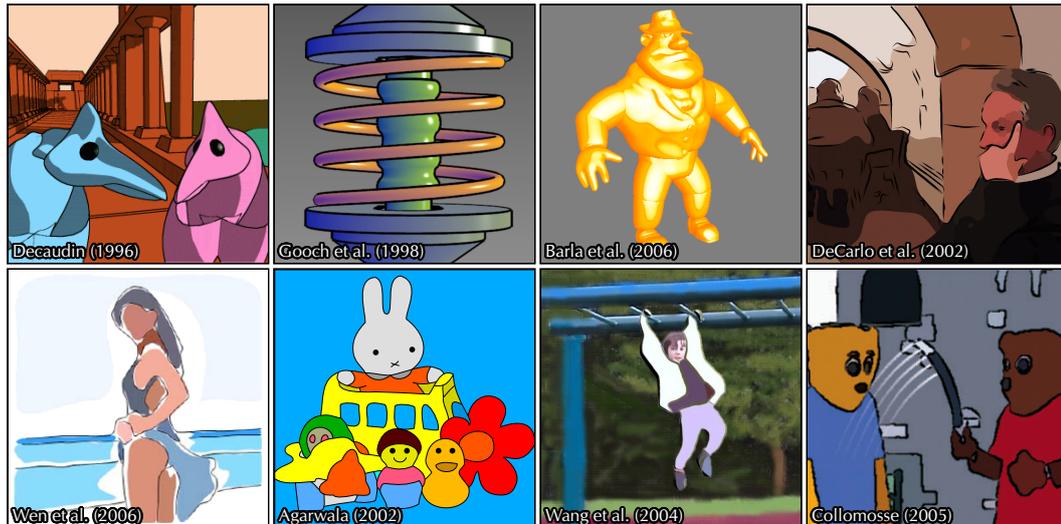
Figure 2.4: Examples of cartoon-like abstraction techniques. © The respective copyright owners.

From images to sketches DeCarlo and Santella (2002) pioneered an approach based on colour segmentation. From a pyramid of segmentations, they select the appropriate level of detail using eye-tracking data. The resulting regions are smoothed and enhanced with abstracted black contours. Wen et al. (2006) extend this idea to use interactive segmentation and a colour shift procedure based on artists' colour choices.

Video cartoons Agarwala (2002) introduced the first semi-automatic system to convert videos to cartoons. A user rotoscopes, or outlines, objects on keyframes, to which Bézier splines are fitted using active contours. These contours are then tracked to the next frame and refined. However, as contours will eventually deviate from their intended position, manual intervention will be necessary to adjust them. Instead, Wang et al. (2004a) segment the video volume overnight and use rotoscoped outlines in keyframes to group sub-volumes into semantic regions. These regions then define smooth trajectories for interpolating outlines between keyframes. A third system, by Collomosse et al. (2005), segments video frames individually and links up segments across frames. The linked regions are then smoothed using interpolating spline surfaces to create 'stroke surfaces' representing the boundaries between objects. This intermediate representation can be used for creating temporally coherent animations, using painterly, sketchy and cartoon rendering styles.

Line drawings

Introduction Line drawings are amongst the most common and concise illustration styles – only a few strokes can convey tone, texture and shape (Cole et al., 2009; Figure 2.5).

Pen & ink drawings Winkenbach and Salesin (1994) did some of the earliest work in non-photorealistic rendering on computer-generated pen-and-ink illustration. They introduced 'stroke textures' as a means to express both tone and texture using line drawings. Without the use of stroke textures, Hertzmann and Zorin (2000) illustrate smooth surfaces by robustly computing visible silhouettes and placing hatch marks directly using a direction field defined on the object surface.
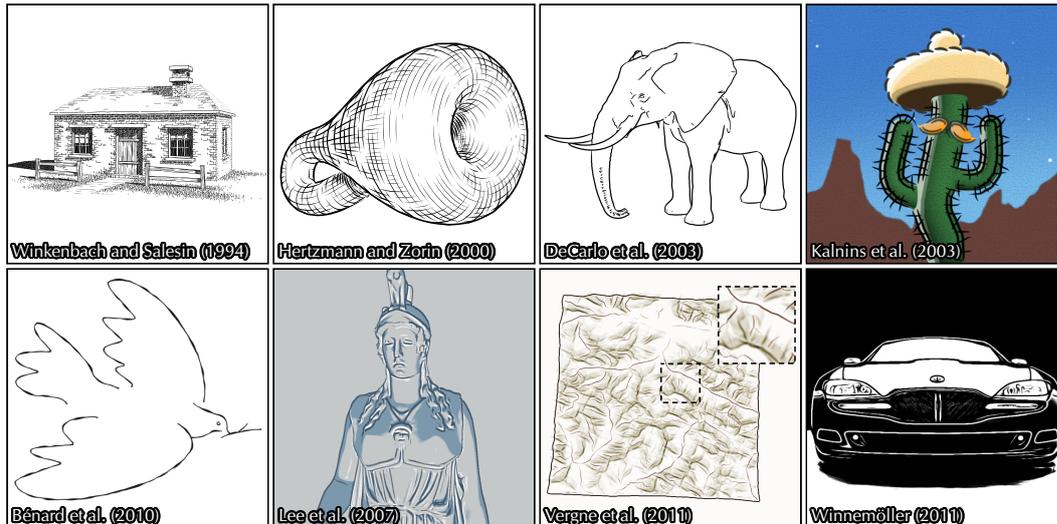
Figure 2.5: Examples of line drawing rendering techniques. © The respective copyright owners.

Contours (or silhouettes) are perhaps the most basic object representation. However, additional cues can be gained from DeCarlo et al.'s suggestive contours (2003) which are lines that are contours in nearby viewpoints. Suggestive contours can be automatically extracted from 3D models – in object- or image-space – and they come quite close to human line drawings (Cole et al., 2008). *Suggestive contours*

Kalnins et al. (2003) describe a technique for rendering coherent stylised silhouettes which are not just straight lines. Their main contribution is a coherent parametrisation of silhouettes that allows strokes to be propagated to the next frame in different ways while maintaining temporal coherence. Bénard et al. (2010) introduce self-similar line artmaps which maintain similar appearance at all zoom levels. This avoids two undesirable artefacts of Kalnins et al.'s approach: sliding stroke textures and stroke texture stretching. Bénard et al. further describe how to generate self-similar line artmaps from a single exemplar. *Stylised silhouettes*

Lee et al. (2007) propose a simple real-time technique that extracts line drawings from a diffusely shaded image. Their technique detects both valleys and ridges in the shaded image to draw both dark lines and highlight lines (which provide additional lighting cues). These lines are then combined with cel shading to capture large-scale tone variations. This technique will be used in Section 5.3.2. *Line drawings via abstracted shading*

Vergne et al. (2011) generalises previous image-space line drawing techniques and introduces a new line-based rendering technique called 'implicit brushes'. It is based on convolving a brush footprint with a feature skeleton that is fitted to surface profiles in image space. This approach is temporally coherent and does not require any temporal feature tracking. *Implicit brushes*

Winnemöller (2011) combines the extended Difference-of-Gaussians (DoG) edges of Winnemöller et al. (2006) with the flow-based DoG edges of Kyprianidis and Döllner (2008). The result is a new operator with large stylistic potential covering effects such as hatching, high-detail artistic thresholding and negative edges. *Extended DoG edges*

2.1.2. Stylisation

Motivation   Just a few pages ago, I defined stylisation as the creation of aesthetic imagery and the explorion of novel artistic techniques. A wide range of techniques have explored this space, but most have concentrated on recreating and automating traditional media such as oil paintings and watercolours.

Structure   This section surveys painterly stylisation techniques proportionally to the attention paid to these techniques in the literature. Hence, this section is largely concerned with stroke-based rendering and will only briefly cover watercolour rendering and more exotic stylisation approaches. I show selected examples in Figures 2.6 and 2.7.

Stroke-based rendering

Pioneering work   Haeberli's pioneering "Paint By Numbers" system (1990) made it possible for users to place brush strokes on a digital canvas simply by clicking. The stroke colour and orientation was then taken from a reference image and orientation field. In contrast, Meier (1996) renders 3D models by attaching particles to objects and projecting them into screen space. These particles are then rendered as brush strokes from back to front, using object geometry and lighting to determine stroke orientation and colour. The result of this process is a temporally coherent video stylisation.

Painterly videos   Litwinowicz (1997) established the core techniques for stroke-based rendering of videos. His system initially places brush strokes in a regular grid and aligns them tangentially to image gradients. In low-gradient regions, stroke orientations can optionally be interpolated from nearby strong gradients using thin-plate splines. For a more hand-painted look, the strokes are perturbed in length, colour and orientation, and drawn in random order. Between frames, the strokes are tracked using optical flow, requiring strokes to be inserted or deleted to maintain a uniform coverage, which can cause strokes to 'pop' (appear suddenly). This technique was also used (with considerable manual input) to create an 8.5 minute painterly video sequence for the 1998 motion picture *What Dreams May Come* – winning the Academy Award for Best Visual Effects (Green et al., 1999).

Improvements   Hertzmann (1998) proposed a multi-resolution approach from coarse to fine brush strokes, which has been widely adopted. He also introduced curved brush strokes based on cubic B-spline brushes that follow gradient lines. This technique is also the initialisation for the video-based technique by Hertzmann and Perlin (2000), which moves and warps strokes over time using optical flow. In subsequent frames, regions of significant change are overpainted, to better approximate the source frame at the cost of sacrificing temporal coherence. Hertzmann also provides a good survey of stroke-based rendering techniques (2003), including connections to other artistic styles such as stippling, mosaics and pen-and-ink drawings.

Temporal coherence   Hays and Essa (2004) contributed two improvements to increase temporal coherence of painterly animations: (1) newly inserted strokes are smoothly faded in and removed ones faded out; and (2) stroke properties such as position and orientation are temporally limited to prevent abrupt changes leading to 'popping' strokes.
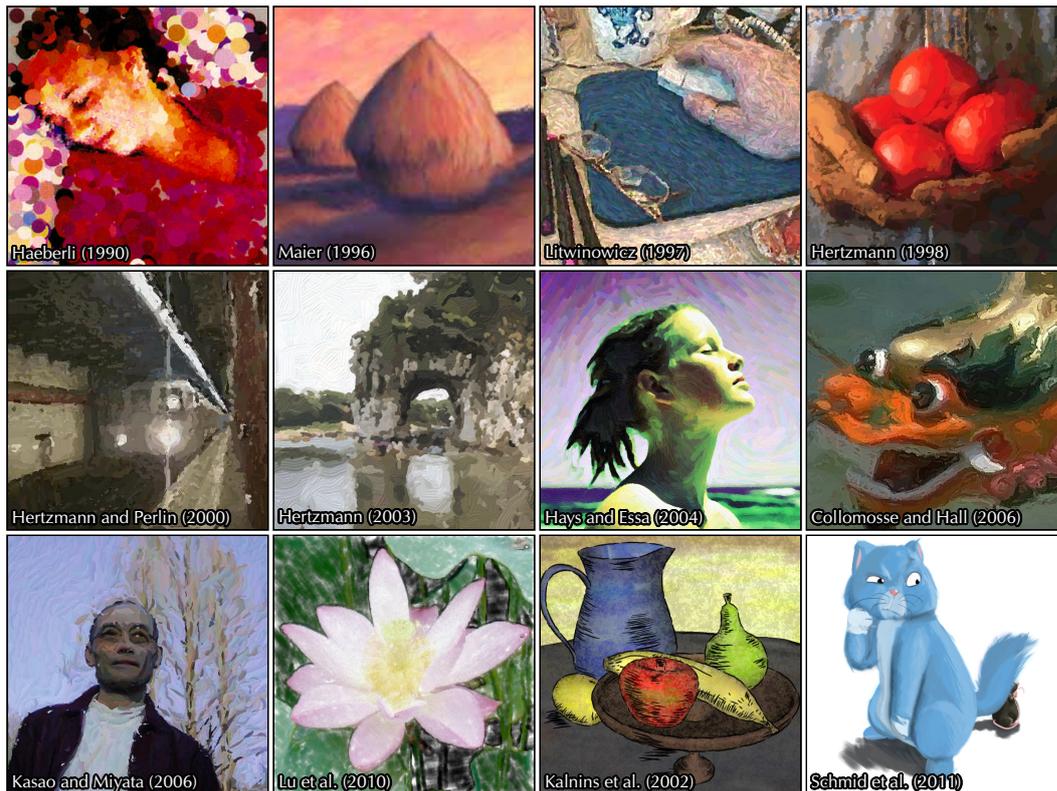
Figure 2.6: Examples of stroke-based rendering techniques. © The respective copyright owners.

Inspired by the way artists strive to capture salient elements of a scene, Collomosse and Hall (2006) proposed a novel painting algorithm that searches for a globally 'optimal' painting in the sense that it preserves salient image detail and attenuates non-salient image regions. Brush strokes are optimised using genetic programming on a compute cluster, and they eventually converge to the optimal painting. Also inspired by real artists, Kasao and Miyata (2006) classify image segments into edge areas, homogeneous areas and contrasting areas, and then stylise them differently. *Inspired by artists*

Lu et al. (2010) introduce a real-time system that unifies the conversion of images, videos and 3D animations into painterly stylisations. Like Hays and Essa (2004), they place strokes across the virtual canvas and move them according to optical flow. They further propose a stochastic approach to maintain uniform coverage of brush strokes by making on a per-pixel level. I discuss their technique in more detail in Section 5.3.3 where I also adapt it to stylise videos with depth. *Real-time stylisation*

Instead of providing a fully automatic system, Kalnins et al.'s 'WYSIWYG NPR' system (2002) lets artists draw directly onto 3D geometry – also in multiple views. Their system then combines the drawn strokes into one model which can then be rendered from any new viewpoint while adapting the number and placement of strokes to maintain a similar look. Schmid et al. (2011) extend this approach in their 'OverCoat' system to let artists use the full 3D space as a canvas. At the heart of their system is proxy geometry which defines an implicit canvas in 3D space. Artists can then draw brush strokes on level sets of the proxy geometry, or using their proposed hair and feather tools. *Tools for artists*
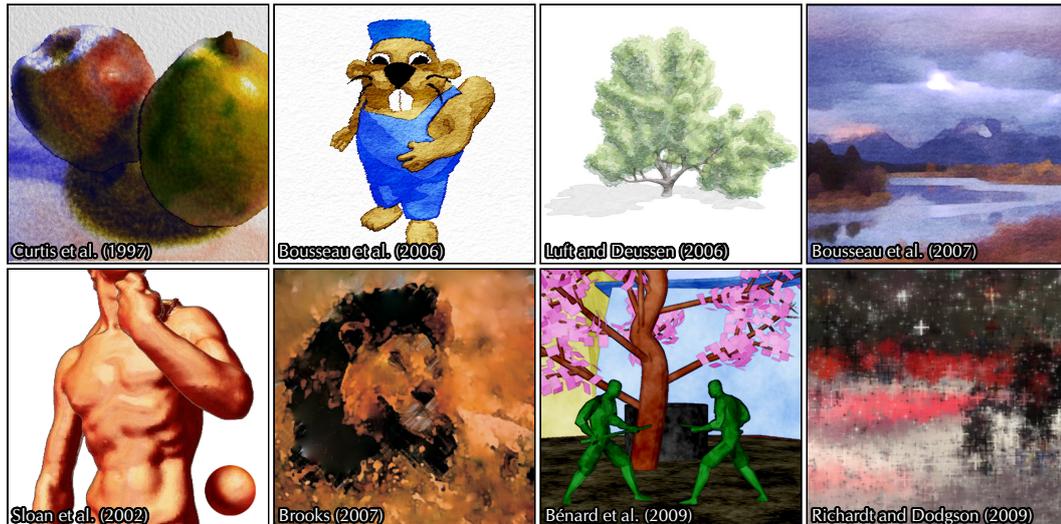
Figure 2.7: Examples of watercolour and other rendering techniques. © The respective copyright owners.

Watercolour and other rendering styles

Watercolour rendering

Watercolour stylisation is much less common in the literature – perhaps due to the increased complexity compared to other techniques. Curtis et al. (1997) simulate watercolour effects using a shallow-water fluid simulation in combination with a pigment compositing model. While this is computationally expensive, it produces realistic results. They also propose semi-automatic 'watercolourisation'. Instead of simulating watercolour effects, Bousseau et al. (2006) recreate them using a number of filters for specific effects such as dry brush, wobbling and edge darkening. Luft and Deussen (2006) present an approach that visually and geometrically abstracts plant models, and renders them using a blurred depth test for increased temporal coherence. Lastly, Bousseau et al. (2007) extended their previous work to video using temporally coherent techniques for shape-abstracting videos and creating the watercolour pigment textures.

Miscellaneous stylisations

A few non-photorealistic rendering techniques defy classification, but are still worth mentioning. For example, Sloan et al. (2001) introduced 'lit spheres' which capture artistic shading models in the form of spherical exemplars. They can easily be applied to objects by using surface normals to index into the lit sphere textures. Brooks (2007) has proposed an approach that mixes two or more artistic media in the same image. Regions in the source image are processed differently depending on their frequency content and finally recombined in the gradient domain. To create temporally coherent stylisations of 3D geometry, Bénard et al. (2009) introduced 'dynamic solid textures' which build on an infinite zoom mechanism that displays the right level of detail regardless of zoom level. Section 6.1 uses their technique to create stereo-coherent watercolour renderings. Lastly, I proposed a video stylisation framework (Richardt and Dodgson, 2009) based on Grundland et al.'s image stylisation framework (2008), which splits the video volume into 3D Voronoi cells. This sparse video representation affords a wide variety of artistic rendering styles which reconstruct stylised video frames from surrounding colour samples.

2.1.3. Stereoscopic non-photorealistic rendering

Stereoscopic images consist of two slightly different 'half-images': one for the left *Introduction* eye and one for the right eye. The binocular disparity between the two half-images provides an important depth cue to the human visual system (see Section 2.2). Naturally, stereoscopic non-photorealistic rendering aims to produce two renditions with appropriate disparity to convey a sense of depth. This is also an important aim of my work and the final results are shown in Section 5.4.

The main difficulty of stereoscopic approaches is the coherence of the two views. If *Stereoscopic* features in one view cannot be matched with corresponding features in the other *coherence* view, the viewer will experience discomfort and depth perception may ultimately break down. In fact, Chapter 6 shows that the coherence between half-images is strongly correlated with stereoscopic viewing comfort. By measuring the coherence, one can therefore predict viewing comfort from an input stereo image.

Bartesaghi et al. (2005) describe a hatching method that uses geometric information *Normal-based* acquired using stereo matching (Section 2.4) or photometric stereo (Section 2.3.1). *hatching* Their approach computes surface normals and principal curvatures, either directly from photometric stereo or from the stereo disparity map and its derivative. The surface normal is then used to express tone (lighting) and the smoothed direction field is used to locally rotate Tonal Art Maps (Praun et al., 2001).

Most work in stereoscopic non-photorealistic rendering has been carried out at the *Work from Vienna* Vienna University of Technology by Marković, Stavrakis and Gelautz in 2004–2008. They considered many rendering styles, including stereoscopic painterly rendering, abstraction and line drawings, and subsequently summarised their observations and advice for creating stereoscopic artwork (Stavrakis and Gelautz, 2005b). Their work also culminated in two PhD dissertations (Marković, 2007; Stavrakis, 2008). The input to all their techniques are self-recorded stereo images from which they then compute disparity maps using stereo matching (Section 2.4) – mostly using their own technique (Bleyer and Gelautz, 2005).

Their first technique is a painterly rendering system (Stavrakis and Gelautz, 2004). *Painterly rendering* They initialise the left view using Hertzmann's technique (1998) and then warp the strokes to the right view using the disparity map. However, some areas may have become disoccluded and need to be filled in again by drawing additional strokes underneath. The result can be viewed with adjustable horizontal displacement which changes the visible parallax. Gelautz et al. (2004) extend this approach by stylising depth layers independently and processing videos frame by frame, without temporal coherence. Stavrakis and Gelautz (2005a) additionally provide an interface to vary the level of detail by hiding layers with smaller brush sizes.

Stavrakis et al. (2005) next propose a simple stereoscopic abstraction method with *Image abstraction* the aim to effectively communicate shape and the distances between objects. As previously, the left view is abstracted first by applying colour-based segmentation on each depth layer. The segments are then warped from the left to the right view and occluded areas are filled in. The last step overlays black outlines where strong edges in the image and disparity map coincide.
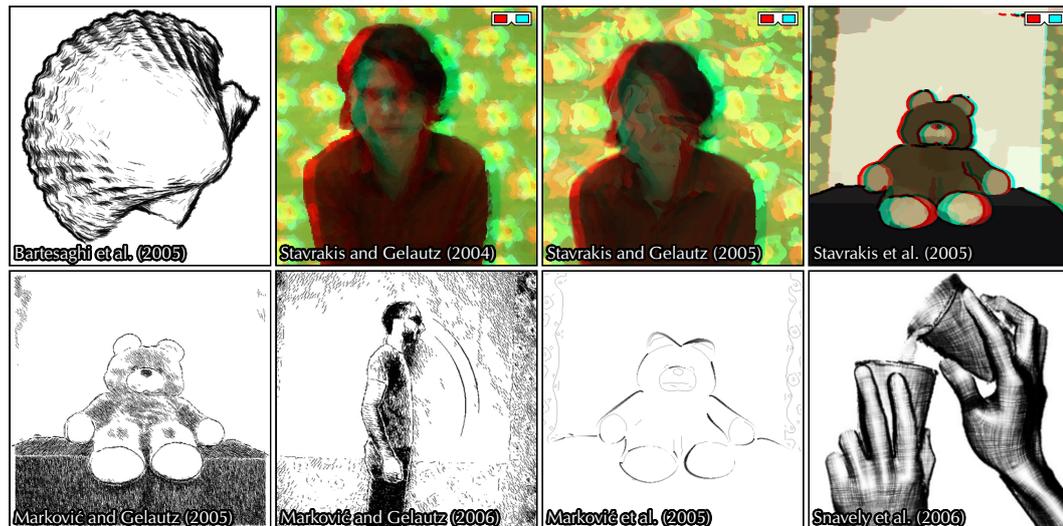
Figure 2.8: Examples of stereoscopic non-photorealistic rendering. © The respective copyright owners.

Drawings & sketches In contrast to their previously described techniques, which produce stereoscopic images, the following work produces only monoscopic results. Drawings are the subject of Marković and Gelautz (2005). A greyscale version of the left half-image determines the placement of strokes and their density using Poisson disc sampling with varying disc radii related to the greyscale value. Individual strokes are oriented along disparity isophotes and important edge features are outlined using the same edge combination approach as in their abstraction work. Marković and Gelautz (2006) additionally depict motion lines and contours extracted from a stereo video. Marković et al. (2005) also create sketches solely from intensity edges and depth discontinuities. They match depth edges to nearby intensity edges in the left image and approximate the resulting lines using Bézier splines – allowing for smoothing and stylisation by varying the width of lines.

Stylising 2.5D video Snavely et al. (2006) stylise video with per-pixel depth information, which they call *2.5D video* and I call *RGBZ video*. This work is also relevant as 2.5D video is an also intermediate result when using stereo video input. After segmenting the video into foreground and background, they fill holes using simple interpolation and smooth each depth map independently using a bilateral filter. Next, they estimate shape correspondence between frames, which enables temporally coherent stylisation. For a hatching style, they fix hatch marks to the surface of objects and track them over time using the shape correspondence. They also create a painterly rendition using Hertzmann's curved brush strokes (1998) with Hays and Essa's temporal coherence improvements (2004). Their results are monoscopic, but extending their technique to stereo videos would result in consistent stereo half-images, as they are based on the same reconstructed geometry.

Inspiration Snavely et al.'s work is a major inspiration for Chapter 4, in which I introduce a prototype RGBZ video camera that efficiently captures high-quality RGBZ videos, and Chapter 5, in which I demonstrate a range of novel video processing effects that are made possible using these high-quality RGBZ videos.

2.2. Human depth perception

The human visual system lets us perceive the world around us in three dimensions. *Introduction*
To achieve this, it combines different types of visual information into a coherent
visual model of the world (Howard and Rogers, 2008). The information that feeds
into this process includes *depth cues* about the relative or absolute depth of objects,
but also our interpretation using prior knowledge of the world and the objects in it.

An understanding of depth perception is important for many tasks in computer *Relevance*
graphics and vision. Computer graphics can benefit by effectively, and correctly,
using depth cues to communicate visual information (Pfautz, 2000). In computer
vision, it is the field of stereo computer vision (Section 2.4) that is inspired the
most by the human perception of depth. In my dissertation, I heavily draw on
depth perception for coherent stereo matching (Chapter 3), stereoscopic rendering
of videos (Section 5.4) and predicting stereoscopic viewing comfort (Chapter 6).

2.2.1. Depth cues

The sources of evidence that are combined to provide depth perception are known *Discovery*
as *depth cues*. Many of them, like linear perspective, and light and shade, were
first discovered by Renaissance painters. Other depth cues were discovered more
recently, and new depth cues are still being found. However, the neural process by
which these cues are combined to create a sense of depth is still not well understood
(Cutting and Vishton, 1995; Ponce and Born, 2008).

There is some disagreement about the definitive list of depth cues and their names *List of depth cues*
in the literature. The following list of 11 depth cues is largely based on the account
by Lipton (1982), with the addition of focal blur, which was proposed more recently.
Depth cues include:

– **Occlusion** or **interposition**
  Objects that occlude other objects are perceived as being closer. This powerful
  cue provides information about the relative depth order of objects. Occlusion is
  widely considered to be the strongest depth cue (Cutting and Vishton, 1995).

– **Linear perspective**
  Parallel lines, such as railroad tracks heading away from the viewer, appear to
  converge with distance and eventually join up at infinity in a vanishing point
  on the horizon. The more the lines converge, the further away they appear.

– **Size cues**
  As objects get further and further away, their projection onto the retina subtends
  a smaller and smaller angle, and they are thus perceived as being further away.
  Prior knowledge about an object also allows us to determine its absolute depth.

– **Aerial perspective**
  As light travels through the atmosphere, it is subject to atmospheric effects, such
  as scattering, which result in distant objects appearing more hazy and blurry.
  For example, this haze makes mountains appear far away.

– **Light and shade**
The visual system assumes that light comes from above, and uses the position of highlights and shadows to infer object properties such as depth and surface relief. If an image is viewed upside down, it is perceived completely differently.

– **Texture gradient**
The level of detail of a texture, for example on a road, reduces with distance from the viewer. This loss of visual detail helps us judge depth (Gibson, 1950).

– **Motion parallax**
When an observer is moving, objects at different depths will move at different relative velocities. For example, when driving in a car, nearby foliage will move rapidly while distant hills will appear stationary (Rogers and Graham, 1979).

– **Focal blur** or **depth of field**
Blurred objects tend to be at a different depth than objects which are in focus. The blur creates an impression of depth, for example in tilt-shift photography, which is also effective even when applied artificially to an image (Mather, 1996).

– **Accommodation**
The lens of the eye changes shape to focus at different distances. This is achieved by contracting and relaxing the intraocular muscles, which the visual system interprets as an absolute depth cue.

– **(Con-)Vergence**
When both eyes look at the same point, the eyes rotate to converge on the point. Exercising the extraocular muscles in this way provides an absolute depth cue.

– **Binocular disparity**
Different images are projected onto the retinas of the two eyes, and the *disparity* between the two views is inversely related to the depth of an object.

Classifications These depth cues are most often grouped into *monocular* and *binocular* depth cues, depending on how many eyes are required for them to function: the first nine depth cues on the list above are monocular and the last two (vergence and binocular disparity) are binocular. Alternatively, depth cues can be split into physiological, or *oculomotor*, depth cues (which are experienced) and psychological depth cues (which are used for inference). The physiological depth cues are motion parallax, accommodation, vergence and binocular disparity.

Combination Combining many depth cues reduces the ambiguity of any individual depth cue and results in a powerful sense of three-dimensionality. Some depth cues also dominate others in certain situations (Cutting and Vishton, 1995). Pfautz (2000) provides the example that "a person threading a needle primarily uses stereo cues to determine the location of the end of the thread and the eye of the needle, and usually brings the objects close to the eyes to increase the accuracy of stereo and oculomotor cues".

Stereopsis As the dominant binocular depth cue, binocular disparity is of particular interest in understanding and creating stereoscopic imagery. It is processed by a visual mechanism called 'stereopsis' which is the focus of the next section.

2.2.2. Stereopsis

> [...] the mind perceives an object of three dimensions by means of the two dissimilar pictures projected by it on the two retinæ [...]
>
> — *Wheatstone (1838)*

First described by Wheatstone in 1838, *stereopsis* is the process in visual perception that fuses the two images projected onto the retinas of the two eyes into a combined sensation of depth. The term 'stereopsis' derives from the Greek στερέος (*stereos*) meaning 'solid' or 'three-dimensional', and ὄψις (*opsis*) meaning 'view' or 'sight'.

History

As a consequence of the horizontal displacement of the two eyes, they see the same scene from two slightly different viewpoints. By looking at a particular point in space, such as the point *F* in Figure 2.9, we line up the fovea (the high-resolution centre of the retina) of both eyes with that point. The projection of other points, such as N in Figure 2.9, onto the retina then might result in different retinal distances to the fovea in the left and right eyes. This difference of $(n-f)-(n'-f')$ is called 'binocular disparity' and is the basis for stereopsis (Ponce and Born, 2008).

Binocular disparity
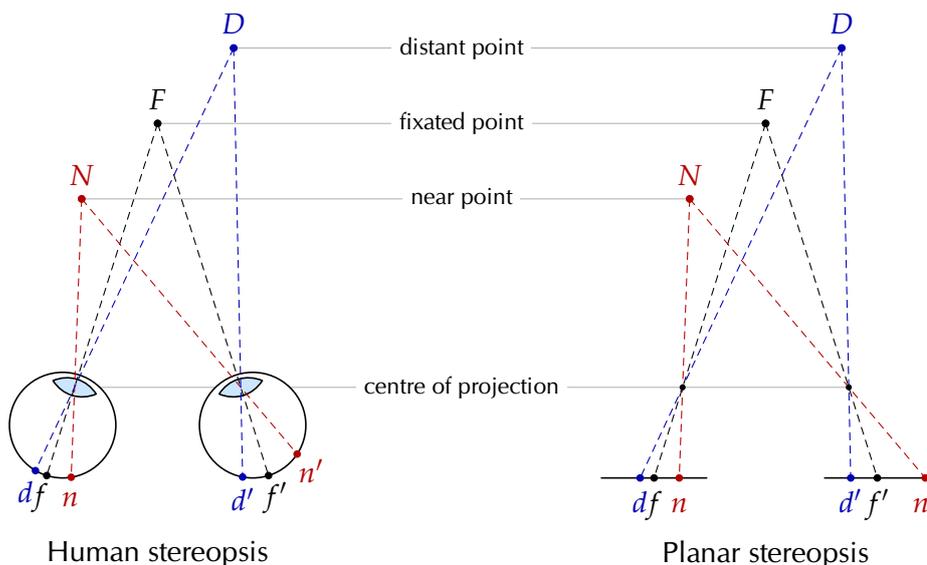
Figure 2.9: Geometry of human stereopsis (left, adapted from Ponce and Born, 2008), and planar stereopsis (right) as used in stereo computer vision and graphics.

The lines connecting nearby points and their retinal projections cross in front of the plane of fixation[1]. This results in 'crossed' disparities which – by convention – are assigned negative values. Conversely, distant points produce 'uncrossed' disparities which have positive values. This explanation is slightly simplified, but sufficient for the purposes of my dissertation. A summary of the geometry and physiology of stereopsis is provided by Ponce and Born (2008), with detailed treatment in textbooks (Tyler, 2004; Howard and Rogers, 2008).

Positive & negative binocular disparity

---

[1] Technically, the 'plane of fixation' is the *horopter* (Ponce and Born, 2008).

Stereoblindness Richards (1970) shows evidence that there are at least three classes of independent disparity detectors – for crossed, zero and uncrossed disparities. His experimental data shows that the probability of each detector to be missing is about 30 per cent. Furthermore, 20 per cent of the experimental population lacked two out of three detectors, and about 2.9 per cent completely lack stereopsis in one hemisphere.

Testing stereopsis A common test for stereopsis was first proposed by Julesz (1964). It is based on a random dot stereogram (a random image of black and white pixels) that is identical for both eyes except for some region that is horizontally displaced (see Figure 2.10). This design eliminates influences from depth cues other than binocular disparity.

negative disparity (in front)   zero disparity (flat)   positive disparity (behind)
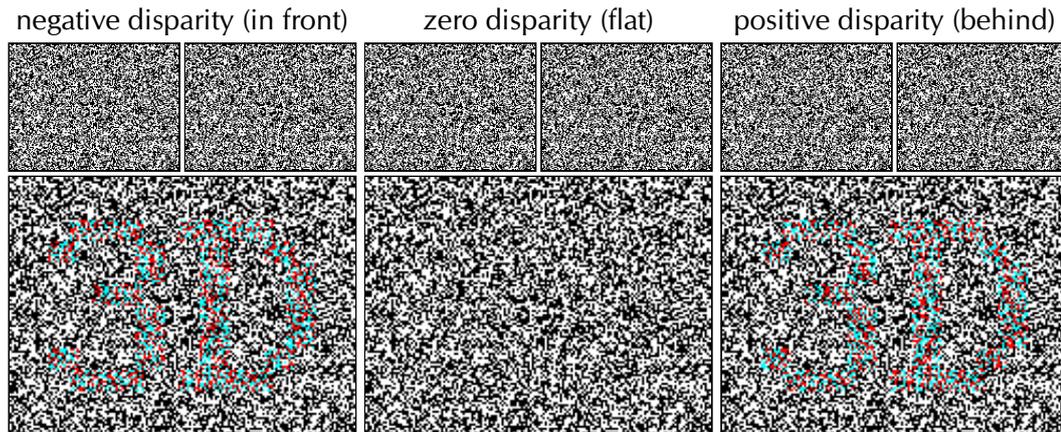
Figure 2.10: Random dot stereograms after Julesz (1964). The upper part of each example shows the stereograms for cross-eyed free viewing, while the bottom shows them as red-cyan anaglyphs.

Zone of comfort For any given distance, the range of binocular disparities resulting in single, fused vision is limited[2]. Even if binocular fusion is possible, it may not be comfortable (Howarth, 2011). The range of permitted disparities is thus limited to the so-called *comfort zone* (Lipton, 1982; Shibata et al., 2011). In addition, Lipton recommends placing important parts of a scene near the plane of fixation.

2.2.3. Stereoscopy

Introduction While stereopsis is the process of interpreting binocular stimuli, *stereoscopy* is the process of presenting binocular stimuli to the two eyes. The term 'stereoscopy' has similar roots to stereopsis, with σκοπέω (*skopeō*) meaning 'to look' or 'to see'. Achieving stereoscopic depth perception requires three levels of binocular vision: simultaneous perception, binocular fusion and finally stereopsis. Therefore, the goal of all stereoscopic viewing techniques is to evoke the depth cues of vergence and binocular disparity by showing different images to both eyes. In the rest of this section, I briefly describe a selection of techniques, from past to present.

Free viewing Stereoscopic images placed side by side can be viewed without the assistance of a viewing device by experienced viewers. There are two *free viewing* approaches: for parallel viewing, the left stereo image is on the left and the eyes converge behind the image plane; and for cross-eyed viewing, the left stereo image is on the right

---

[2] The zone of single binocular vision is known as *Panum's fusional area* (Panum, 1858).

and the eyes converge in front of the image plane. In both cases, the image is still focused on, resulting in an unpleasant vergence–accommodation conflict.

The first dedicated viewing device was the *stereoscope*. In 1833, Wheatstone invented the mirror stereoscope which he describes in his seminal work (1838) – years before photography was practical. It uses two mirrors to reflect each eye's view 90 degrees to either side where the stereo half-images are affixed. Brewster (1856) improved on this design by using two lenses in his enclosed stereoscope, which enlarge the inserted *stereograph*. Stereoscopes were hugely popular between the 1860s and the 1920s – particularly in Britain and the United States – as they provided a life-like depiction of scenes, and hundreds of millions of stereographs were printed.

Stereoscope

Anaglyph images use complementary colour filters, such as red and cyan or green, to present a different image to each eye. This approach was first described by Rollmann (1853) and it led to the first wave of stereoscopic cinema with several short films released in 1922–1941. Its key benefit was and is that it can be used with existing media, that is using the same film projectors, the same digital displays and even on printed paper. However, by showing differently coloured images to each eye, anaglyph images often cause strong visual discomfort after longer exposure. In this dissertation, I show all stereoscopic images as red-cyan anaglyphs, because they provide the best printed results without requiring free viewing. Most anaglyph images are indicated by small red-cyan glasses: ▰▱.

Anaglyph

For the red-cyan anaglyph glasses to work correctly, that is to cleanly separate the red and cyan channels, anaglyph images need to use the right shades of red and cyan. This is more difficult to ensure for printers, which often reproduce colours inaccurately, leading to a degraded stereo viewing experience. For this reason, I recommend to view the anaglyph images in my dissertation on a digital display.

Disclaimer

Full colour reproduction is restored with polarisers: the two views are orthogonally polarised and polarising glasses only let the correctly-polarised light through while blocking the orthogonal direction. Two competing systems exist: linear and circular polarisation. Linear polarisation with two film projectors – one for each view – fuelled a second wave of stereoscopic cinema in the 1950s and is still used in IMAX 3D projection today. More recently, circular polarisation systems were introduced, which allow viewers to tilt their head without breaking the stereo perception. Together with digital projection, this technique is at the heart of the third wave of stereoscopic cinema which started around 2005. It is also used in Section 6.4.

Polarisation

Stereoscopic images can also be shown alternately with synchronised active shutter glasses that quickly alternate making one eye's glass dark and the other transparent. At the time of writing, the most widespread equipment is Nvidia's '3D Vision' kit comprising glasses and an infrared transmitter for synchronisation.

Shutter glasses

Numerous other stereoscopic approaches exist, which I will not describe in detail: head-mounted displays, for example, simply mount two displays in front of the eyes; autostereoscopic approaches do not require any glasses (Dodgson, 2005); holographic techniques can record multiple static views; volumetric displays are physically three-dimensional; and the so-called 'wigglegram' quickly alternates the stereo half-images, which induces motion parallax instead of binocular disparity.

Assorted techniques

2.3. Capturing dynamic geometry at video rates

Introduction Recovering the 3D geometry of objects is a classic problem in computer vision. Many solutions – commonly known as *3D scanners* – have been proposed to capture geometry from one or more images, often with the support of additional equipment such as lights, projectors or sensors. The final result is a description of geometry in terms of the distance, depth, disparity or surface normal of surface points.

Aside on depth versus distance In this dissertation, I follow specific definitions of the terms 'distance' and 'depth'. I use 'distance' to denote the Euclidean distance (also known as 'range') between a 3D point and the camera's centre of projection, whereas 'depth' refers only to the distance along the viewing direction of the camera. So, for example, points at constant distance form a sphere, whereas points at constant depth form a plane.

Motivation This section studies existing geometry capture methods to evaluate their suitability for creating RGBZ videos – videos with depth. The capture approaches hence need to be able to densely capture dynamic geometry at video frame rates of at least 10 Hz. This requirement excludes approaches that only produce sparse depth data or are too slow, such as contact-based scanners, LIDAR[3] and other laser scanners. This leaves four approaches which are surveyed next: photometric stereo, stereo matching, active stereo and time-of-flight sensors (Sections 2.3.1 to 2.3.4).

2.3.1. Photometric stereo

Description The idea of photometric stereo is to compute dense normal maps from several images of an object under different lighting conditions. This technique – also known as *shape-from-shading* – originates in work by Horn (1970) who considered the case of shape recovery from a single image. This was extended by Woodham (1980) to exploit images of multiple known lighting directions under the assumption that objects are Lambertian reflectors. Zhang et al. (1999) provide a survey of the more recent shape-from-shading literature.

Applications Photometric stereo is often used in offline scenarios to acquire high-quality dense normal maps, for example for non-photorealistic rendering (Toler-Franklin et al., 2007). By using multiple LEDs and quickly cycling through them, photometric stereo can also be applied at video frame rates (Wang et al., 2010). However, object motion between frames can lead to normal artefacts. To avoid artefacts, Malzbender et al. (2006) use a high-speed video camera which allows them to record 8 lighting directions at 60 Hz. These papers' applications to NPR are discussed in Section 5.3.1.

2.3.2. Stereo matching

Description The computer vision version of human stereopsis (Section 2.2.2) is known as *stereo matching* or *stereo correspondence*. This approach computes the disparity between corresponding points in stereo half-images, which can then be converted to depth. Section 2.4 explains this approach in more detail and also describes the typical components of stereo matching techniques. *Multi-view stereo* techniques use more than two cameras, which increases the computational complexity accordingly.

---

[3] 'LIDAR' is an acronym for 'LIght Detection And Ranging'.

2.3.3. Active stereo

Active stereo techniques combine a camera with a projector which casts known light patterns into the scene. This approach is also known as *structured-light scanning*, as the projected light is used to solve the correspondence problem of stereo matching. Using suitable patterns, the mapping from each projector pixel onto the camera's imaging sensor can be worked out, and the original world point can be triangulated. Lanman and Taubin (2009) organised a course on how to build your own scanner.

Description

In November 2010, Microsoft released the Microsoft Kinect (see Figure 2.11), which is the first mass-market product to combine an infrared-based active stereo system with a colour video camera in a single case for a competitive price. The depth sensor operates in infrared light to avoid interference with the scene which is captured by the colour video camera. I use a Kinect sensor in Chapter 4.

Microsoft Kinect

2.3.4. Time-of-flight cameras

Time-of-flight cameras record depth maps by measuring the time it takes a light signal to travel from the camera to an object and back. Two approaches are in use for measuring the time-of-flight of light. The first approach uses a fast optical shutter in front of the image sensor which opens and closes in synchrony with the light pulses that are sent out. The second approach works by modulating the outgoing light and measuring the phase shift of the received signal. Examples of each type are the 3DV ZCam[4] and the MESA Imaging SwissRanger 4000 (Figure 2.11). Current models cannot record colour images, but only an infrared intensity image.

Description

Time-of-flight cameras have many industrial applications, for example in robotics and automobiles, and they are also increasingly used in computer graphics (Kolb et al., 2010). However, the Microsoft Kinect is likely to replace time-of-flight cameras – at least in graphics and vision applications – as it is considerably cheaper.

Applications
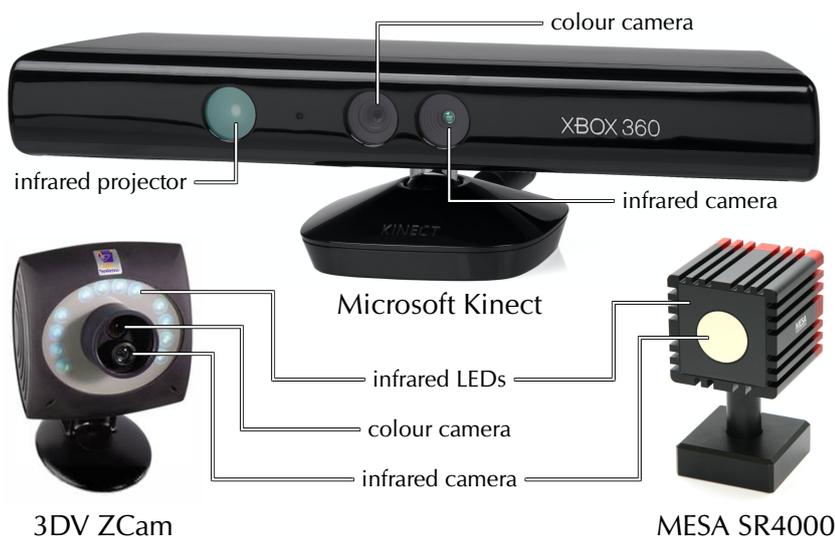
Figure 2.11: Three commercial depth sensors and their components (not to scale).

---

[4] In March 2009, Microsoft acquired the vendor 3DV Systems and later discontinued the ZCam.

2.3.5. Analysis of strengths and weaknesses

Introduction The four described approaches use very different means to recover scene geometry. To find the optimal approach for a particular application, one therefore needs to weigh the pros and cons of each approach. The major advantages and disadvantages of the four approaches are summarised in Table 2.1.

Approach	Advantages	Disadvantages
Photometric stereo (2.3.1)	– high-quality per-pixel surface normals – easy to handle high resolutions	– need several light directions per frame – difficulties with shadows & highlights – need to integrate normals to get depth
Stereo matching (2.3.2)	– passive: no additional light required – based on stereopsis (Section 2.2.2) – well established in literature	– repetitive textures cause ambiguities – poor performance near weak textures
Active stereo (2.3.3)	– independent of scene texture – good depth accuracy	– visible light interferes with scene – projecting multiple patterns is slow
Time-of-flight cameras (2.3.4)	– independent of scene texture – no scene interference	– low spatial resolution – fairly noisy depth data

Table 2.1: Comparison of dynamic geometry capturing approaches.

Requirements This dissertation aims to show that RGBZ videos – which are colour videos with depth – enable the creation of advanced video processing effects that are unfeasible from a colour video alone. To achieve this goal, RGBZ videos hence need to be:

- **clean:** the colour video should be free of artefacts, the depth data free of noise;

- **dense:** every pixel should have an associated depth value;

- **video-rate:** depth and colour data for every RGBZ video frame;

- **synchronised:** no offset in time between colour and depth data;

- **registered:** corresponding depth and colour edges should overlap; and

- **plausible:** geometric accuracy is not required for many applications.

Optimal techniques There is no commercially-available hardware that records RGBZ videos directly. Therefore, I need to build on and extend one of the four mentioned approaches for dynamic geometry capture. In Chapter 3, I extend a stereo matching technique to work on stereo videos, and improve the coherence of the computed disparity maps, particularly in the presence of image noise. However, the depth data is still not of sufficient quality to proceed. In a new attempt, I use time-of-flight cameras in Chapter 4, because their depth maps are independent of scene texture which is a major weakness of stereo matching techniques.

2.4. Taxonomy of stereo correspondence techniques

Stereo vision is one of the most active fields in computer vision – hundreds of stereo matching techniques have been proposed. To organise, compare and evaluate all of them in a meaningful way, Scharstein and Szeliski (2002) have proposed a taxonomy of stereo correspondence algorithms and provide evaluation datasets with results collected and ranked on a website. According to their taxonomy, algorithms generally consist of the following four steps:

Stereo taxonomy

1. matching cost computation;

2. cost (support) aggregation;

3. disparity computation/optimisation; and

4. disparity refinement.

Decomposing a stereo matching technique into these steps helps in comparing the performance of the individual components, although some techniques only use a subset. The following introduces key concepts common to all stereo matching techniques.

The input to all stereo correspondence (or matching) techniques is a single stereo image, which consists of a pair of 'half-images': one for the left view and one for the right view. Scharstein and Szeliski assume that the stereo cameras are calibrated and the half-images rectified, which means that any 3D point is projected onto the same scan-line in both half-images (Hartley and Zisserman, 2004, section 11.12).

Stereo images

Rectified images significantly reduce the complexity of the correspondence problem. Instead of having to search the entire image for a corresponding point (a 2D search), the corresponding point is now constrained to lie on a given scan-line (a 1D search). The displacement along the scan-line between corresponding points in the two half-images is known as their *disparity*.

Disparity

Dense stereo correspondence algorithms ultimately compute a *disparity map* $d(\mathbf{p})$, which assigns a scalar disparity to each pixel $\mathbf{p} = (x, y)$ in a reference image. By convention, the left half-image is the reference image. The corresponding pixel $\overline{\mathbf{p}}$ in the right half-image is then specified by

Disparity map

$$\overline{\mathbf{p}} = (x - d(\mathbf{p}), y). \tag{2.1}$$

Fundamental to stereo matching is also the concept of a *cost space*, which assigns a cost $C(\mathbf{p}, d)$ to all pixels $\mathbf{p}$ and disparity hypotheses d. For any particular pixel, different disparities generally result in different costs. Lower costs indicate better matches depending on how well each disparity explains the observed image evidence. Many stereo matching techniques compute an initial cost space in step 1, refine it by aggregating support in step 2, find the optimal disparity from the cost space in step 3 and refine it in step 4.

Cost space

The four steps of the stereo taxonomy are described in detail in Sections 2.4.1 to 2.4.4, and Scharstein and Szeliski's evaluation methodology is outlined in Section 2.4.5.

Structure of this section

2.4.1. Matching cost computation

Introduction The first step computes the initial cost space $C(\mathbf{p}, d)$. In most algorithms, this initial cost matches individual pixels and it is determined by comparing corresponding pixels. The costs are then aggregated over a neighbourhood in the next step.

Absolute & squared differences The most basic costs are the absolute difference (AD) and squared difference (SD) between pixel values in the left and right half-images, L and R:

$$C_{\mathrm{AD}}(\mathbf{p}, d) = |L_{\mathbf{p}} - R_{\overline{\mathbf{p}}}| \text{ and} \tag{2.2}$$

$$C_{\mathrm{SD}}(\mathbf{p}, d) = (L_{\mathbf{p}} - R_{\overline{\mathbf{p}}})^2, \tag{2.3}$$

where $I_{\mathbf{p}}$ is the value of pixel $\mathbf{p}$ in an image I. When working on colour images, the differences are generally computed per colour component and summed.

Cost truncation For pixel values that are sufficiently different, it is often unimportant how different they are. A useful extension is thus to truncate (or saturate) costs above some threshold τ. The truncated variants of AD and SD are known as TAD and TSD:

$$C_{\mathrm{TAD}}(\mathbf{p}, d) = \min(\tau, |L_{\mathbf{p}} - R_{\overline{\mathbf{p}}}|) \tag{2.4}$$

$$C_{\mathrm{TSD}}(\mathbf{p}, d) = \min(\tau, (L_{\mathbf{p}} - R_{\overline{\mathbf{p}}})^2). \tag{2.5}$$

Cross-correlation Zero-mean normalised cross-correlation (ZNCC) is another frequently used matching cost. It blurs the boundary between cost computation and aggregation, as it correlates (zero-mean normalised) windows of pixels instead of individual pixels:

$$C_{\mathrm{ZNCC}}(\mathbf{p}, d) = \frac{\sum_{\mathbf{q} \in N_{\mathbf{p}}} (L_{\mathbf{q}} - \overline{L_{\mathbf{p}}}) \cdot (R_{\overline{\mathbf{q}}} - \overline{R_{\overline{\mathbf{p}}}})}{\sqrt{\sum_{\mathbf{q} \in N_{\mathbf{p}}} (L_{\mathbf{q}} - \overline{L_{\mathbf{p}}})^2 \cdot \sum_{\mathbf{q} \in N_{\mathbf{p}}} (R_{\overline{\mathbf{q}}} - \overline{R_{\overline{\mathbf{p}}}})^2}}, \tag{2.6}$$

where $\overline{I_{\mathbf{p}}}$ is the mean value of pixels in a square window $N_{\mathbf{p}}$ centred on $\mathbf{p}$:

$$\overline{I_{\mathbf{p}}} = \frac{1}{|N_{\mathbf{p}}|} \cdot \sum_{\mathbf{q} \in N_{\mathbf{p}}} I_{\mathbf{q}}. \tag{2.7}$$

Usage in this dissertation The stereo matching techniques I propose in Chapter 3 use absolute differences without truncation. Furthermore, Chapter 6 predicts stereoscopic viewing comfort using zero-mean normalised cross-correlation as the basis of a computational model of human depth perception.

2.4.2. Cost aggregation

Introduction Pixel-wise matching costs produce very noisy disparity maps, and so costs are typically aggregated over a larger support area. This step essentially combines the pixel-wise matching cost C of the previous step into a window-based matching cost C'. The assumption is that nearby and/or similar pixels have the same disparity, and their costs are hence aggregated to strengthen the correct disparity hypothesis.

Box filter (square window)   Shiftable windows
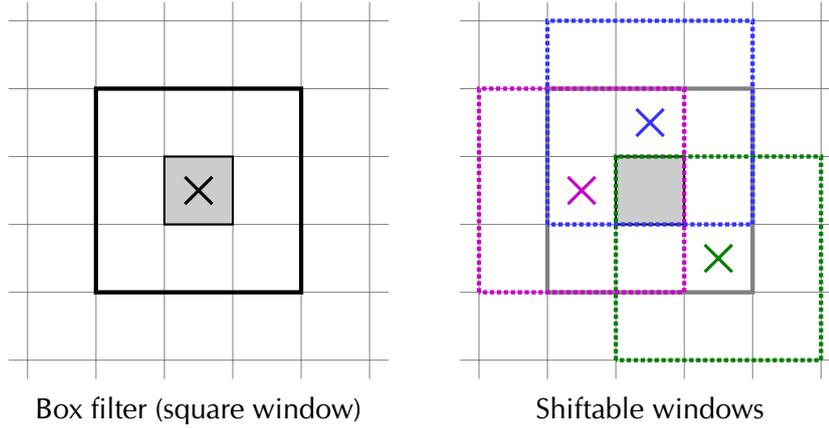
Figure 2.12: Comparison of the box filter and shiftable windows cost aggregation techniques.
Left: The box filter accumulates costs inside the 3×3 window centred on the shaded pixel.
Right: The box filter is shifted across pixels in the solid grey window, and the minimum cost is taken. Three dotted windows are shown centred on different pixels (indicated by crosses).

The simplest cost aggregation technique is the box filter, which sums costs within a *Box filter* square window of a given size around each pixel:

$$C'_{\text{box}}(\mathbf{p}, d) = \sum_{\mathbf{q} \in N_{\mathbf{p}}} C(\mathbf{q}, d). \tag{2.8}$$

The box filter (illustrated in Figure 2.12, left) generally produces a smoother disparity map, but it struggles at depth edges, where the support window straddles both background and foreground. In this case, the costs of both fore- and background will be mixed up, resulting in the thickening of image structures.

This problem can be addressed by shifting the window, so that in the 'edge' case, *Shiftable windows* the window would be entirely in the same region as the pixel under consideration:

$$C'_{\text{shift}}(\mathbf{p}, d) = \min_{\mathbf{r} \in N_{\mathbf{p}}} \sum_{\mathbf{q} \in N_{\mathbf{r}}} C(\mathbf{q}, d). \tag{2.9}$$

However, both the box filter and its shiftable variant cannot handle fine detail in the images, as the level of detail is inevitably limited by their window size.

Yoon and Kweon (2006) proposed a more flexible technique using adaptive support *Adaptive* weights, which are computed using the distance and similarity between pixels. This *support weights* chapter heavily draws on this approach, which is explained in detail in Section 3.1.

2.4.3. Disparity optimisation

Following cost aggregation, the next step finds the optimal disparity for all pixels *Winner-take-all* given the cost space C'. Selecting the best disparity in terms of cost defines the so-called winner-take-all (WTA) technique:

$$d(\mathbf{p}) = \arg\min_{d} C'(\mathbf{p}, d). \tag{2.10}$$

This is very fast, but the downside is that disparities are optimised for each pixel independently, not taking into account the disparities of surrounding pixels.

Local versus global methods When the steps described so far are used together, the result is a so-called *local* stereo correspondence technique, as each pixel's disparity is the result of computations that only depend on a local neighbourhood around each pixel and not on image data further away. *Global* techniques optimise across the entire disparity map. This enables them, for example, to propagate edge information across weakly textured regions in the image. They often also enforce smoothness constraints that restrict the disparity map to be piecewise smooth.

Global optimisation The highest ranked optimisation techniques express the stereo matching problem as a Markov Random Field (MRF), which they solve iteratively. The main competing approaches are graph cuts (Boykov et al., 2001; Kolmogorov and Zabih, 2001) and belief propagation (Sun et al., 2003; Felzenszwalb and Huttenlocher, 2006). These techniques produce excellent results, but are computationally expensive and difficult to implement efficiently, particularly on GPUs. Nevertheless, efficient GPU implementations have been proposed for (single-label) graph cuts (Vineet and Narayanan, 2008) and belief propagation (Liang et al., 2009).

Usage in this dissertation For well-conditioned stereo images, local stereo matching approaches with winner-take-all produce results of almost similar quality to global techniques. However, local techniques are more amenable to efficient GPU implementations, as will be demonstrated in the next chapter.

2.4.4. Disparity refinement

Introduction The computed disparity maps can be rough and noisy, so a range of refinement techniques have been proposed to further improve disparity maps. The quantised nature of disparity maps can be ameliorated using sub-pixel refinement techniques. Many errors in disparity maps are caused by occlusion, and techniques like the left-right check detect inconsistent pixels and invalidate them. General post-processing also includes filling invalidated regions and 'clean up' using median filters.

Sub-pixel refinement The disparity map is usually limited to a discrete set of disparities, often integers. To reduce the quantisation steps between adjacent disparities, Yang et al. (2007) propose a sub-pixel refinement step. For each pixel, they fit a quadratic polynomial to the three cost values around the lowest cost, $C'(\mathbf{p}, d(\mathbf{p}) + \{-1, 0, 1\})$, and find the minimum to determine the refined sub-pixel disparity:

$$d'(\mathbf{p}) = d(\mathbf{p}) - \frac{1}{2} \cdot \frac{C'(\mathbf{p}, d(\mathbf{p}) + 1) - C'(\mathbf{p}, d(\mathbf{p}) - 1)}{C'(\mathbf{p}, d(\mathbf{p}) + 1) - 2 \cdot C'(\mathbf{p}, d(\mathbf{p})) + C'(\mathbf{p}, d(\mathbf{p}) - 1)}. \quad (2.11)$$

Left-right check The left-right check (LRC) is a popular technique for identifying occluded and other inconsistent pixels in disparity maps (Egnal and Wildes, 2002). It works on two disparity maps: the left-to-right disparity map d_L and the right-to-left disparity map d_R. Since both disparity maps should be 'inverses' of each other, the disparities of corresponding pixels should sum to zero. A pixel $d_L(\mathbf{p})$ in the left-to-right disparity map is hence considered consistent if this sum falls below a threshold T_{LRC} (which is usually set to $T_{\text{LRC}} = 1$):

$$\left| d_L(\mathbf{p}) + d_R(\overline{\mathbf{p}}) \right| < T_{\text{LRC}}. \quad (2.12)$$

2.4.5. Middlebury stereo benchmark

In addition to proposing a taxonomy of stereo matching approaches, Scharstein and Szeliski discuss different approaches to evaluate stereo matching techniques using images with ground truth disparity maps (in their section 5). The approach that stood the test of time is the percentage of 'bad pixels', which are those that deviate more than a given threshold from the correct disparities. Scharstein and Szeliski also contribute stereo images with ground truth disparities: first piecewise planar (2002) and later also more complex scenes (2003). *Quantitative evaluation*

Perhaps the most useful contribution of Scharstein and Szeliski is the Middlebury stereo website[5]. It lists a total of 114 stereo matching techniques as of November 2011, and evaluates their performance on four stereo images with ground truth disparity maps. Disparity maps for the four images are uploaded by paper authors and then automatically evaluated and listed on the website. *Middlebury stereo website*

Each image is assessed, and ranked accordingly, in 3 categories: the nonoccluded pixels, all pixels and pixels near depth discontinuities (called *nonocc*, *all* and *disc*, respectively). The average of the 12 ranks for a technique determines its average rank, shown in the green column of Figure 2.13. Techniques are also sorted by average rank, which means that the relative order of techniques can change when new techniques are added to the table. *Ranking techniques*

Error Threshold = 1		Sort by nonocc			Sort by all			Sort by disc				Sort by disc			
Algorithm	Avg. Rank	**Tsukuba** ground truth			**Venus** ground truth			**Teddy** ground truth			**Cones** ground truth			Average percent of bad pixels (explanation)
		nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	
ADCensus [94]	6.2	1.07 13	1.48 10	5.73 15	0.09 2	0.25 7	1.15 2	4.10 5	6.22 3	10.9 5	2.42 3	7.25 5	6.95 4	3.97
AdaptingBP [17]	7.8	1.11 16	1.37 6	5.79 16	0.10 3	0.21 4	1.44 4	4.22 7	7.06 6	11.8 8	2.48 5	7.92 10	7.32 8	4.23
CoopRegion [41]	7.8	0.87 3	1.16 1	4.61 2	0.11 4	0.21 3	1.54 6	5.16 15	8.31 11	13.0 12	2.79 14	7.18 4	8.01 18	4.41
DoubleBP [35]	10.4	0.88 5	1.29 3	4.76 5	0.13 7	0.45 18	1.87 11	3.53 4	8.30 10	9.63 3	2.90 18	8.78 26	7.79 15	4.19
RDP [102]	10.8	0.97 8	1.39 7	5.00 7	0.21 22	0.38 15	1.89 12	4.84 9	9.94 17	12.6 10	2.53 6	7.69 7	7.38 9	4.57
OutlierConf [42]	11.3	0.88 4	1.43 9	4.74 4	0.18 15	0.26 9	2.40 19	5.01 11	9.12 14	12.8 11	2.78 13	8.57 21	6.99 5	4.60
SubPixDoubleBP [30]	15.5	1.24 24	1.76 26	5.98 20	0.12 6	0.46 20	1.74 9	3.45 3	8.38 12	10.0 4	2.93 20	8.73 25	7.91 17	4.39
SurfaceStereo [79]	15.9	1.28 29	1.65 18	6.78 34	0.19 17	0.28 10	2.61 26	3.12 3	5.10 1	8.65 1	2.89 17	7.95 12	8.26 24	4.06

Figure 2.13: The top stereo matching techniques shown on the Middlebury stereo benchmark website.

---

[5] http://vision.middlebury.edu/stereo/eval/

2.5. A brief introduction to bilateral filtering

Motivation The bilateral filter is a common edge-preserving smoothing filter. In contrast to the Gaussian filter, which blurs image content across edges, the bilateral filter preserves image edges. This is achieved by adapting the filter kernel to the image content, making it a non-linear filter. As the bilateral filter is both versatile and conceptually simple, it is widely used in computer graphics and computer vision.

History The bilateral filter has been discovered independently at least three times. Aurich and Weule (1995) first proposed the bilateral filter as a non-linear Gaussian filter for "edge-preserving diffusion" and also demonstrated its edge-sharping qualities. The authors further considered filter chains with varying filter parameters, to improve the smoothing effect, and proposed a cross-bilateral extension (Section 2.5.2) where filter stages use the previous filtering result to smooth the original input image. In concurrent work[6], Smith and Brady (1997) introduce the bilateral filter as part of their SUSAN low-level vision framework for "structure preserving noise reduction". In 1998, Tomasi and Manduchi gave the filter its current name and showed the first colour filtering results. They also remarked that filtering all colour components jointly, for example in the CIELAB colour space, produces visually better results than filtering the red, green and blue components independently. Paris et al. (2008) provide "a gentle introduction to bilateral filtering and its applications"[7].

| original image | Gaussian blur | bilateral filter |
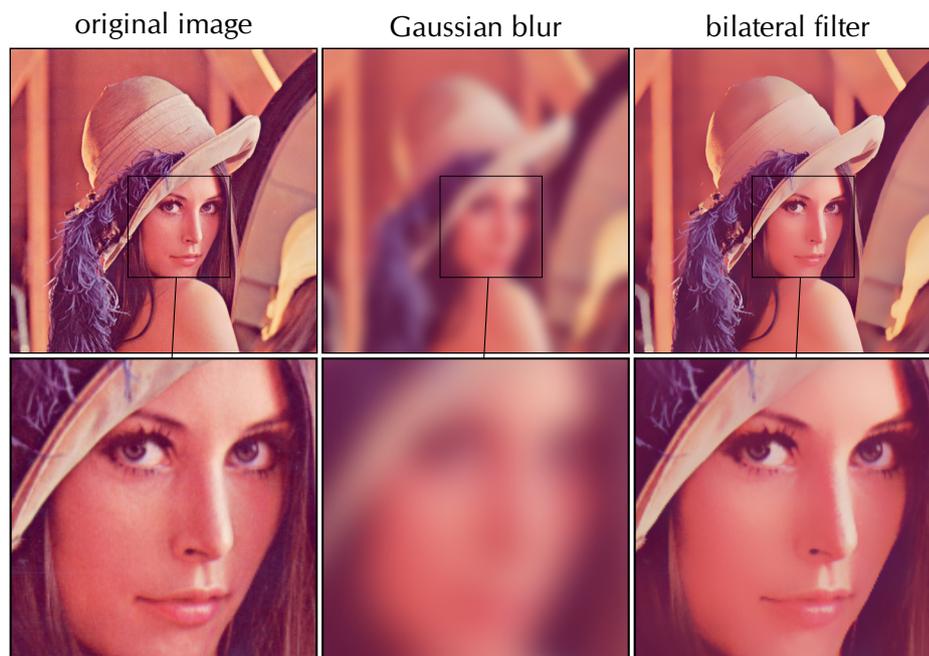
Figure 2.14: Example results of a Gaussian blur ($\sigma = 10$) and a bilateral filter ($\sigma_s = 10$, $\sigma_r = 20/256$) applied to the Lena image. Notice how the bilateral filter removes fine detail such as image noise and hair strands while preserving strong edges in the image.

---

[6] Although Smith and Brady's work was submitted earlier (3 May 1993 versus 31 March 1995), it was not accepted until five months after Aurich and Weule's work (23 October versus 31 May 1995) and not published until 16 months later (January 1997 versus September 1995).

[7] A slightly extended version of their SIGGRAPH 2008 course was published as Paris et al. (2009).

2.5.1. Formulation of the bilateral filter

Like convolution-based filters, such as the Gaussian blur, the bilateral filter replaces each pixel with a linear combination of other pixel values. While convolution-based filters apply the same mask of weights, known as a *kernel*, to each pixel that is filtered, the bilateral filter adapts the kernel weights to the surrounding pixels according to the distance and similarity of the pixels. *Outline*

Specifically, for a pixel $\mathbf{p}$ in an image I, the filtered pixel value $I'_{\mathbf{p}}$ is calculated as *Generalised formulation*

$$I'_{\mathbf{p}} = \frac{\sum\limits_{\mathbf{q} \in N_{\mathbf{p}}} f(\|\mathbf{p} - \mathbf{q}\|) \cdot g(\|I_{\mathbf{p}} - I_{\mathbf{q}}\|) \cdot I_{\mathbf{q}}}{\sum\limits_{\mathbf{q} \in N_{\mathbf{p}}} f(\|\mathbf{p} - \mathbf{q}\|) \cdot g(\|I_{\mathbf{p}} - I_{\mathbf{q}}\|)}, \tag{2.13}$$

where the pixel $\mathbf{q}$ ranges over $N_{\mathbf{p}}$, a set of pixels which conceptually is the set of all pixels $\mathcal{P}$, and the corresponding pixel values $I_{\mathbf{q}}$ are weighted by f and g, which are functions of the distance $\|\mathbf{p} - \mathbf{q}\|$ and difference in value $\|I_{\mathbf{p}} - I_{\mathbf{q}}\|$ between pixels, respectively. The sum of all weights in the denominator normalises the linear combination of pixel values $I_{\mathbf{q}}$ in the numerator.

In computer graphics, the bilateral filter is most commonly used with Gaussian weighting functions $f(x) = G_{\sigma_s}(x)$ and $g(x) = G_{\sigma_r}(x)$, where $G_{\sigma}(x) = e^{-x^2/2\sigma^2}$. This approach provides parameters σ_s and σ_r for adjusting the spatial and range bandwidths, respectively. Rewriting Equation 2.13 accordingly results in *Gaussian formulation*

$$I'_{\mathbf{p}} = \frac{\sum\limits_{\mathbf{q} \in N_{\mathbf{p}}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) \cdot G_{\sigma_r}(\|I_{\mathbf{p}} - I_{\mathbf{q}}\|) \cdot I_{\mathbf{q}}}{\sum\limits_{\mathbf{q} \in N_{\mathbf{p}}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) \cdot G_{\sigma_r}(\|I_{\mathbf{p}} - I_{\mathbf{q}}\|)}. \tag{2.14}$$

Adams et al. (2009) introduce a simplification of the filter notation by representing pixel values as homogeneous quantities, such as $(r, g, b, 1)$ instead of (r, g, b), and filtering the homogeneous coordinate like the others. This notation will be assumed from this point on, as it eliminates the usual division by the sum of weights: *Homogeneous notation*

$$I'_{\mathbf{p}} = \sum_{\mathbf{q} \in N_{\mathbf{p}}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) \cdot G_{\sigma_r}(\|I_{\mathbf{p}} - I_{\mathbf{q}}\|) \cdot I_{\mathbf{q}}. \tag{2.15}$$

Naïve implementations of the bilateral filter evaluate Equation 2.13 directly for each pixel in the image. This approach uses two nested loops: the outer loops over $\mathcal{P}$, the set of all pixels in the image I, and the inner loops over $N_{\mathbf{p}}$, the set of pixels that determine the new pixel value $I'_{\mathbf{p}}$. Assuming that pixel weights can be computed in constant time, the total time complexity is $\mathcal{O}(|\mathcal{P}| \cdot |N_{\mathbf{p}}|)$. *Computational complexity*

Assuming that $N_{\mathbf{p}} = \mathcal{P}$ means that every pixel influences every other's pixel value. However, in practice, this is unnecessary, as the weights given to pixels far from $\mathbf{p}$ will be small, and their contribution to the filtered pixel value will be negligible (Paris et al., 2008). For Gaussian weighting functions, the filter kernel $N_{\mathbf{p}}$ is thus commonly limited to a square of 'radius' $2\sigma_s$ or $3\sigma_s$ centred on $\mathbf{p}$. This reduces the computational complexity from $\mathcal{O}(|\mathcal{P}|^2)$ to $\mathcal{O}(|\mathcal{P}| \cdot \sigma_s^2)$. *Kernel truncation*

2.5.2. Cross-bilateral filtering

Formulation A powerful extension of the bilateral filter is the *cross-* or *joint-bilateral* filter proposed concurrently by Eisemann and Durand and Petschnigg et al. in 2004. The key idea is to filter the input image *I* using another image *E* from which to extract the edge information:

$$I'_{\mathbf{p}} = \sum_{\mathbf{q} \in N_{\mathbf{p}}} G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) \cdot G_{\sigma_r}(\|E_{\mathbf{p}} - E_{\mathbf{q}}\|) \cdot I_{\mathbf{q}}. \tag{2.16}$$

Example One application of this filter is in 'flash/no-flash' photography, where a noisy no-flash photograph can be filtered using the clean, but flat-coloured flash photograph of the same scene, to produce a denoised version of the no-flash photograph.

Usage in this dissertation The cross-bilateral filter is used extensively in this dissertation. In the next chapter, I extend it into a cost aggregation technique for stereo matching (Section 3.1.2). In addition, Chapter 4 builds a geometry fill-in procedure (Section 4.2) as well as a spatiotemporal filtering technique (Section 4.3) on top of the cross-bilateral filter.

2.5.3. Acceleration approaches

Brief overview Naïve implementations of the bilateral filter are very slow, so many acceleration approaches have been proposed. Durand and Dorsey's layered approximation (2002) achieves a significant speedup, but is not faithful to the full-kernel filtering result. A separable implementation has been proposed by Pham and van Vliet (2005), but it can cause axis-aligned filtering artefacts. Weiss' technique (2006) only supports spatial box-filters rather than Gaussian weights, and Yang et al.'s constant-time bilateral filtering (2009) does not generalise well to higher dimensions.

High-dimensional Gaussian filtering The bilateral filter can be reinterpreted as a Gaussian filter in a higher-dimensional space (Barash, 2002). For this, the pixel coordinates $\mathbf{p} = (x, y)$ are augmented by the pixel values $I_{\mathbf{p}} = (r, g, b)$ to yield new 5D pixel coordinates $\mathbf{p}' = (\mathbf{p}, I_{\mathbf{p}}) = (x, y, r, g, b)$. This higher-dimensional space is then filtered using a 5D axis-aligned Gaussian with standard deviations $\sigma = (\sigma_s, \sigma_s, \sigma_r, \sigma_r, \sigma_r)$:

$$I'_{\mathbf{p}} = \sum_{\mathbf{q}' \in N_{\mathbf{p}'}} G_{\sigma}(\|\mathbf{p}' - \mathbf{q}'\|) \cdot I_{\mathbf{q}}. \tag{2.17}$$

This turns the bilateral filter into an almost entirely linear filter. The only non-linear element is the division by the homogeneous coordinate (as per homogeneous filter notation by Adams et al., 2009).

Bilateral grid As the Gaussian blur is separable, it can also be implemented efficiently. This is exploited by the bilateral grid (Paris and Durand, 2009), which stores a coarsely quantised version of the 3D space for greyscale filtering (no full colour). It is also amenable to real-time GPU implementation, as demonstrated by Chen et al. (2007). The bilateral grid is described in detail in Section 3.2.1 and used shortly afterwards.

Sparse approaches More recently, Adams et al. (2009, 2010) introduced two sets of data structures and algorithms to sparsely represent the higher-dimensional space, and thus allow full-colour filtering (as well as other applications). These techniques use Gaussian KD-trees (2009) and permutohedral lattices (2010), respectively.

Coherent depth from stereo matching

3

> *This chapter presents research that has been published and demonstrated at the European Conference on Computer Vision 2010 in Crete, Greece (Richardt et al., 2010b). Douglas Orr implemented the stereo matching infrastructure (Section 2.4), and Ian Davies created the ground truth stereo videos (Section 3.4.2).*

A popular passive geometry capturing approach is (multi-view) stereo (Section 2.3). *Introduction* Passive geometry acquisition has the primary advantage that a scene is observed without throwing light into the scene, which is energy efficient, but also does not cause interference which would disturb people during recordings. Also, humans have evolved to perceive depth from binocular stimuli (Section 2.2), which has motivated research in computer stereo vision for decades.

However, most research in stereo vision has concentrated on still images, and not *Motivation* video sequences and their associated problems. Applying any stereo technique on a sequence of frames most likely results in a sequence of temporally incoherent disparity maps. This incoherence is predominantly caused by noise in the video. In some form or other, continuity between video frames needs to be exploited to ensure temporal coherence of the disparity maps.

The naïve approach to incorporate temporal information into stereo matching is to *Early work* extend the spatial smoothness constraint into time. Leung et al. (2004) propose such an energy minimisation approach and a method to solve it quickly (though approximately) using iterated dynamic programming. The results of their implementation are not convincing, as they are very coarse. Gong (2006) takes another approach by extending a standard local stereo matching technique (as described in Section 2.4) to compute costs for different 'disparity flow' hypotheses. This improves coherence for videos that have little motion between frames – their default settings limit motion to four pixels spatially and one disparity level per frame.

Markov Random Fields (MRFs) provide a clean foundation to optimise disparity *MRF-based global* maps in both space and time. Williams et al. (2005) and Isard and MacCormick *optimisation* (2006) propose spatiotemporal extensions that enforce piecewise smoothness to compute coherent disparity maps, and also optimise for occlusions and motion

estimation, respectively. While these approaches are conceptually attractive, they are extremely computationally expensive due to their dense connectivity and large label space: Williams et al.'s technique takes 6 minutes per 320×240 frame, and Isard and MacCormick's technique 5 seconds per 50×40 frame. This becomes impractical for real-world video resolutions and larger disparity ranges.

Local stereo matching The techniques presented in this chapter take a different approach than these global stereo correspondence techniques. They are based on Yoon and Kweon's adaptive support weights (2006) which aggregate evidence only over a finite window size. The effectiveness of their technique is due to aggregation of support over large window sizes as well as weights that adapt according to similarity and proximity to the central pixel in the support window. The results are good, but the algorithm is slow, taking about one minute for the basic *Tsukuba* stereo image (384×288 pixels). My approach improves on their performance.

Structure of this chapter This chapter starts with an explanation of Yoon and Kweon's technique (Section 3.1), and then continues by rewriting their technique as a *dual-cross-bilateral filter* with Gaussian weights. This allows me to approximate it using the bilateral grid to achieve a speedup of about 200× and improve its accuracy using a dichromatic approach (Section 3.2). Results for these still-image techniques show good real-time performance on the Middlebury benchmark (Section 3.3). I finally present a spatio-temporal extension that incorporates temporal evidence in real time (Section 3.4).

Published resources I believe in making source code and datasets publicly available for other researchers to reproduce my results and compare their techniques to my work more easily. The project website[8] thus contains a wealth of supplementary material: the source code of all implemented techniques, the five synthetic stereo videos we created, and supplementary videos showing the benefit of the spatiotemporal stereo matching approach can all be found there and are linked to throughout this chapter.

---

[8] http://richardt.name/dcbgrid/

3.1. Adaptive support weights as a bilateral filter

Yoon and Kweon (2006) describe a simple, but effective cost aggregation method *Introduction* for local stereo matching (see Section 2.4.2 for this stage in the stereo taxonomy). *& motivation* The aggregated cost is determined by a weighted linear combination of the costs of neighbouring pixels with the same disparity; the weights of pixels in this window depend on their similarity and proximity to the central pixel. The key assumption is that similar and close-by pixels likely have similar disparities.

This aggregation is in effect a *dual-cross-bilateral filter* applied to the cost volume: *Reformulation* *cross-bilateral* because it smoothes the cost volume based on the stereo half-images, and *dual* as it preserves edges in both stereo half-images. Following this insight, Yoon and Kweon's technique is reformulated as dual-cross-bilateral (DCB) cost aggregation with Gaussian weights (Section 3.1.2). This prepares the ground for speeding up the cost aggregation using the bilateral grid (Section 3.2).

3.1.1. Adaptive support weights

Simple cost aggregation approaches, such as shiftable windows (Section 2.4.2), *Introduction* aggregate costs over a fixed-size square window of pixels. All pixels carry the same weight in the cost aggregation regardless of the image content in those regions, although they may have different colours or be at entirely different depths. The aggregation simply averages over all pixels. The result is that fine detail is lost, as the window size determines the finest level of detail that can be matched.

Yoon and Kweon (2006) proposed a simple solution to this problem: *adaptive support* *Idea* *weights*. This is motivated by the *Gestalt* theory of perceptual grouping (Wertheimer, 1923; Todorović, 2008), which states that individual objects are grouped together by proximity as well as similarity. Yoon and Kweon additionally advocate the use of a large support window to provide sufficient spatial support for a particular disparity hypothesis. At the same time, the larger window size does not limit the level of detail, as the adaptive support weights respect image edges.

The support weight $w(\mathbf{p}, \mathbf{q})$ between two pixels $\mathbf{p}$ and $\mathbf{q}$ is calculated by *Formulation*

$$w(\mathbf{p}, \mathbf{q}) = \exp\left(-\frac{\Delta E^*_{ab}(\mathbf{p}, \mathbf{q})}{\gamma_c}\right) \cdot \exp\left(-\frac{\|\mathbf{p} - \mathbf{q}\|}{\gamma_p}\right), \tag{3.1}$$

where ΔE^*_{ab} is the Euclidean distance between pixel colours in the CIELAB colour space, and the parameters γ_c and γ_p control grouping by similarity and proximity, respectively. Yoon and Kweon use default values of $\gamma_c = 5$ and $\gamma_p = 17.5$. Figure 3.1 illustrates some examples of adaptive support weights.

Using the stereo notation of Section 2.4, the aggregated cost space C' at pixel $\mathbf{p}$ for *Cost aggregation* a disparity hypothesis d is calculated by weighting each pixel in a window around $\mathbf{p}$ as per

$$C'(\mathbf{p}, d) = \frac{1}{k} \cdot \sum_{\mathbf{q} \in N_{\mathbf{p}}} w(\mathbf{p}, \mathbf{q}) \cdot w(\overline{\mathbf{p}}, \overline{\mathbf{q}}) \cdot C(\mathbf{q}, d), \tag{3.2}$$

where $k = \sum_{\mathbf{q} \in N_{\mathbf{p}}} w(\mathbf{p}, \mathbf{q}) \cdot w(\overline{\mathbf{p}}, \overline{\mathbf{q}})$ is the normalisation quotient and $N_{\mathbf{p}}$ the set of pixels in the support window of size 35×35, which is centred on the pixel $\mathbf{p}$.
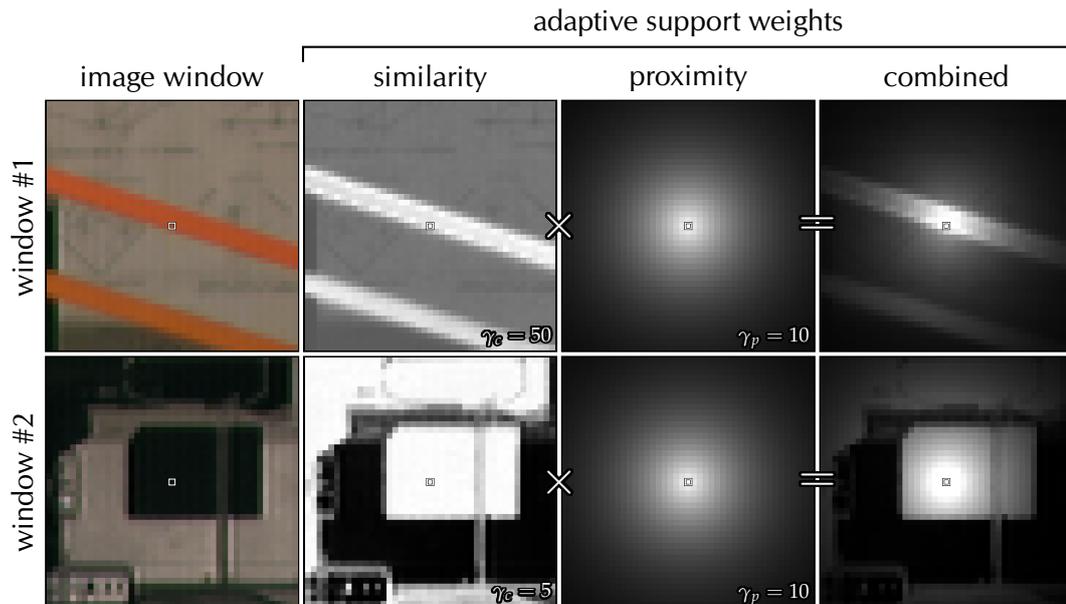
Figure 3.1: Computation of adaptive support windows (Yoon and Kweon, 2006) for two pixels from the *Tsukuba* stereo image. The pixel position is marked by a rectangle in the centre of the windows. Lighter shades indicate higher weights.

Stereo pipeline For the remaining stages in Scharstein and Szeliski's taxonomy in Section 2.4, Yoon and Kweon use truncated absolute differences to compute costs, and then optimise disparities using winner-take-all, without any disparity refinement. The techniques described in this chapter further apply Yang et al.'s sub-pixel refinement step (2007), as described in Section 2.4.4. This reduces quantisation artefacts in the disparity maps without negatively influencing error metrics.

Implementation & discrepancies My straightforward GPU implementation produces comparable results to Yoon and Kweon's publicly-available, CPU-based implementation[9], while being about $25\times$ faster than their reported run times. However, neither implementation achieves the results reported in the original paper. These differences appear to be caused by post-processing the raw disparity maps, which is not discussed in the paper.

Post-processing To compare the different techniques in this chapter fairly, only GPU techniques are compared to other GPU techniques. Furthermore, all techniques share the same post-processing consisting of the following four components:

1. The left-right check invalidates inconsistent pixels (with threshold $T_{\text{LRC}} = 1$).

2. Invalid pixels are then filled using a median filter, if they have at least four valid pixels in their 8-connected neighbourhood.

3. Runs of invalid pixels along a scan-line are detected, and filled using the lower of the two disparities found just before and after the invalid segment.

4. A final median filter reduces noise and removes outliers.

---

9 On the Middlebury stereo website: http://vision.middlebury.edu/stereo/code/.

3.1.2. Dual-cross-bilateral cost aggregation

The cross- or joint-bilateral filter is a variant of the bilateral filter (Section 2.5) which smoothes an image with respect to edges in a different image. Yoon and Kweon's adaptive support weights approach is similar to this in that it smoothes the cost space while preserving edges in both stereo half-images. In the bilateral filtering framework, this kind of filter could be called *dual-cross-bilateral* (DCB), as it filters the cost space *cross-bilaterally* with respect to two images (*dual* filtering). *Parallels*

Yoon and Kweon's approach can be reformulated using Gaussian weights – the de facto standard in bilateral filtering. This turns Equation 3.1 into *Reformulated weights*

$$w(\mathbf{p}, \mathbf{q}) = G_{\sigma_r}(\Delta E_{ab}^*(\mathbf{p}, \mathbf{q})) \cdot \sqrt{G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|)}, \tag{3.3}$$

where σ_r and σ_s are similarity and proximity parameters, and $G_\sigma(x) = e^{-x^2/2\sigma^2}$ is the unnormalised Gaussian centred on zero, with standard deviation σ. The square root is applied to the second factor in Equation 3.3 so that $w(\mathbf{p}, \mathbf{q}) \cdot w(\overline{\mathbf{p}}, \overline{\mathbf{q}})$ includes the proximity weight exactly once.

The aggregation remains unchanged from Equation 3.2, resulting in *Reformulated cost aggregation*

$$C'(\mathbf{p}, d) = \frac{1}{k} \cdot \sum_{\mathbf{q} \in N_\mathbf{p}} w(\mathbf{p}, \mathbf{q}) \cdot w(\overline{\mathbf{p}}, \overline{\mathbf{q}}) \cdot C(\mathbf{q}, d) \tag{3.2}$$

$$= \frac{1}{k} \cdot \sum_{\mathbf{q} \in N_\mathbf{p}} G_{\sigma_r}(\Delta E_{ab}^*(\mathbf{p}, \mathbf{q})) \cdot G_{\sigma_r}(\Delta E_{ab}^*(\overline{\mathbf{p}}, \overline{\mathbf{q}})) \cdot G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|) \cdot C(\mathbf{q}, d), \tag{3.4}$$

computed within the same window of 35×35 pixels as per the original approach. Experimentally, the parameter values $\sigma_r = 10$ and $\sigma_s = 10$ produce good results.

In general, filter windows are often truncated at 2 sigma (Section 2.5.1), resulting in a support window size $(4 \cdot \sigma_s + 1) \times (4 \cdot \sigma_s + 1)$. If σ_s is too small, the support window fails to aggregate sufficient support, but if it is too large, it will negatively impact the run time which grows quadratically in σ_s. On the other hand, the value of σ_r controls the range of pixel values that contribute and their weight: if it is too small, only few pixels with very similar colours will contribute; and if it is too large, too many pixels will contribute, some perhaps unintentionally. *Meaning of σ_s & σ_r*

3.2. Approximation using the bilateral grid

Introduction The previous section reformulated Yoon and Kweon's adaptive support weights as dual-cross-bilateral cost aggregation with Gaussian weights. This section shows how the bilateral grid (Section 3.2.1) can be extended to accelerate dual-cross-bilateral cost aggregation. The result is the *dual-cross-bilateral (DCB) grid* (Section 3.2.2). As the DCB grid uses only greyscale inputs, it performs worse than the full-kernel DCB approach. To recover some of the accuracy, Section 3.2.3 proposes a dichromatic approach which incorporates a second colour axis into the DCB grid. Section 3.3 evaluates all techniques in terms of Middlebury accuracy and run times.

3.2.1. The bilateral grid

Acceleration approaches Section 2.5.3 discussed some approaches for speeding up the bilateral filter. From these, the bilateral grid (Chen et al., 2007; Paris and Durand, 2009) is the best fit for a GPU-based implementation of Equation 3.4, and is hence used. The bilateral grid counter-intuitively also runs faster and uses less memory as standard deviations increase, as data is sub-sampled proportionally, which is useful for accumulating support over large support windows.

1D example Consider the example of a greyscale image $I(x, y)$. The bilateral grid embeds it in a 3D space: 2D for spatial coordinates and 1D for pixel values. Each pixel (x, y) is mapped to $(x, y, I(x, y))$ in the bilateral grid Γ. The 1D example in Figure 3.2 illustrates the use of the bilateral grid in three steps.

1. Grid creation

Create All grid voxels (x, y, c) are first zeroed using $\Gamma(x, y, c) = (0, 0)$. All pixels $I(x, y)$ are then accumulated into the grid Γ using

$$\Gamma\left(\left[\frac{x}{s_s}\right], \left[\frac{y}{s_s}\right], \left[\frac{I(x,y)}{s_r}\right]\right) \mathrel{+}= (I(x,y), 1)\,, \tag{3.5}$$

where $[\,\cdot\,]$ is the rounding operator, and s_s and s_r are the spatial and range sampling rates, which are set to σ_s and σ_r, respectively. Note that the pixel values and the number of pixels are accumulated using homogeneous coordinates, which make it easy to compute weighted averages in the grid slicing stage.

2. Grid processing

Process The grid is now convolved with a Gaussian filter, of standard deviation σ_s and σ_r along the space and range dimensions, respectively. As the previous step has already sub-sampled the data accordingly, we only need to convolve each dimension with a 5-tap 1D Gaussian kernel with $\sigma = 1$.

3. Grid slicing

Slice The result is now extracted by accessing the grid coordinates $(x/s_s, y/s_s, I(x,y)/s_r)$ using trilinear interpolation, and dividing the homogeneous vector to access the actual filtered values.
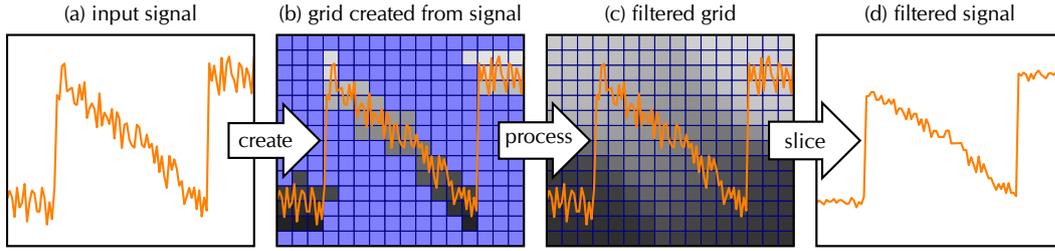
Figure 3.2: Illustration of 1D bilateral filtering using the bilateral grid: the signal (a) is embedded in the grid (b), which is processed (c) and sliced to obtain the filtered signal (d). Please see Section 3.2.1 for details. Adapted from Chen et al. (2007).

3.2.2. Extending the bilateral grid to the dual-cross-bilateral grid

Chen et al. (2007) show that the bilateral grid can also be used for cross-bilateral filtering. This is achieved by using different images for indexing into and storing values in the grid: the edge image $E(x,y)$ determines grid coordinates and the other image $I(x,y)$ determines the stored values to be filtered:

Cross-bilateral filtering using the bilateral grid

$$\Gamma\left(\left[\frac{x}{s_s}\right], \left[\frac{y}{s_s}\right], \left[\frac{E(x,y)}{s_r}\right]\right) \mathrel{+}= (I(x,y), 1).\qquad(3.6)$$

The grid processing remains the same, and the slicing stage accesses the grid accordingly at $(x/s_s, y/s_s, E(x,y)/s_r)$.

Recall that the *dual-cross-bilateral* cost aggregation smoothes the cost space while preserving edges in both stereo half-images. To implement this using the bilateral grid, it needs to be extended to take into account both input images as edge images when calculating grid coordinates, and to accumulate cost space values instead of pixel values. This extension is called the *dual-cross-bilateral (DCB) grid.*

The DCB grid

For a pixel $\mathbf{p} = (x,y)$ in the left image, and its corresponding pixel $\overline{\mathbf{p}} = (x-d, y)$ in the right image, the DCB grid for a disparity d is created using

DCB grid creation

$$\Gamma_d\left(\left[\frac{x}{\sigma_s}\right], \left[\frac{y}{\sigma_s}\right], \left[\frac{L_L^\star(\mathbf{p})}{\sigma_r}\right], \left[\frac{L_R^\star(\overline{\mathbf{p}})}{\sigma_r}\right]\right) \mathrel{+}= (C(\mathbf{p}, d), 1),\qquad(3.7)$$

where the subscripts L and R indicate the left and right images, respectively.

Instead of image intensities, as per Chen et al., this formulation uses the lightness component $L^\star$ of the CIELAB colour space which is perceptually more uniform and hence more closely models how humans perceive greyscale images. However, this also degrades accuracy compared to a full-colour approach such as the full-kernel DCB aggregation. The trade-off between the number of colour dimensions and the corresponding memory requirements for the bilateral grid is discussed in more detail in the next section.

Limitation to lightness

Finally, the result of slicing the DCB grid is the aggregated cost

DCB grid slicing

$$C'(\mathbf{p}, d) = \Gamma_d\left(\frac{x}{\sigma_s}, \frac{y}{\sigma_s}, \frac{L_L^\star(\mathbf{p})}{\sigma_r}, \frac{L_R^\star(\overline{\mathbf{p}})}{\sigma_r}\right).\qquad(3.8)$$

Implementation For each disparity d, the corresponding DCB grid Γ_d is a four-dimensional array of two floating-point numbers. To efficiently implement all grid processing operations, each grid is first flattened into two dimensions. All flattened grids are then tiled into a single 2D texture to have fast random access to the grid data and to exploit texture filtering hardware. This is crucial to efficiently perform the grid slicing step.

Grid flattening The bilateral grid downsamples each dimension by a factor σ. For a value $v \in [0, x]$, the downsampled range of values are therefore the integers $0, 1, \ldots, \lceil x/\sigma \rceil$, a total of $\lceil x/\sigma \rceil + 1$ values. An image of size $w \times h$ and with lightness $L^\star \in [0, 100]$ then has a DCB grid of the dimensions $d_w \times d_h \times d_c \times d_c$, where $d_w = \lceil (w-1)/\sigma_s \rceil + 1$, $d_h = \lceil (h-1)/\sigma_s \rceil + 1$ and $d_c = \lceil 100/\sigma_c \rceil + 1$. As illustrated in Figure 3.3, the 4D DCB grid is then split into a 2D layout of $d_c \times d_c$ components of size $d_w \times d_h$ each, resulting in a flattened size of $(d_w \cdot d_c) \times (d_h \cdot d_c)$.
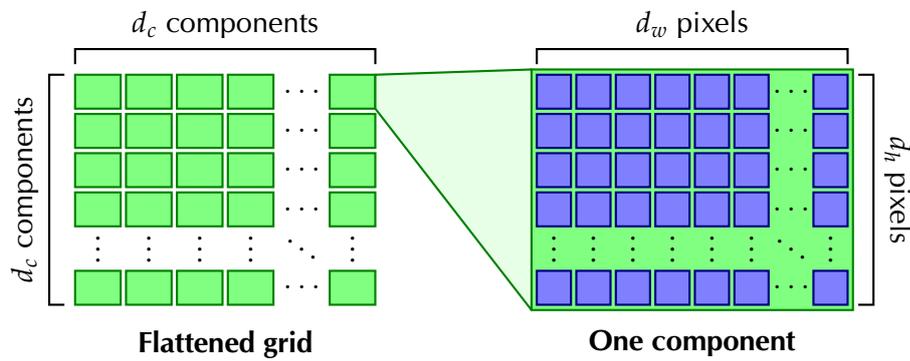
Figure 3.3: Illustration of flattening the four-dimensional DCB grid into two dimensions.

Quadrilinear interpolation The quadrilinear interpolation of the slicing stage can be efficiently implemented by using hardware-accelerated bilinear texture filtering to fetch the values stored at the four surrounding $d_w \times d_h$ components. Within each component, bilinear texture reads are used, and the four resulting values are bilinearly interpolated.

Grid tiling All the flattened grids, one for each disparity, are then tiled into a 2D texture of `float2`s. Given a particular number of grids, a rectangular layout of grids is looked for as it makes optimal use of graphics memory without wasting memory. To find a rectangular tiling, all combinations of numbers of rows and columns are examined, starting from a square configuration, and stopping at the first tiling that fits into the maximum texture size of 8192×8192 pixels. In the case that no valid rectangular tiling exists (see example below), the grids are laid out in reading order instead, fitting as many in horizontally as possible. Note that this may result in texture space being allocated which goes unused.

Example The *Teddy* stereo image has a resolution of 450×375 pixels and 60 disparity levels. Assuming default parameters of $\sigma_s = 10$ and $\sigma_r = 10$, each DCB grid has a size of $46 \times 38 \times 11 \times 11$, or $(46 \cdot 11) \times (38 \cdot 11) = 506 \times 418$ when flattened. A rectangular tiling of 10×6 exists, resulting in a $(6 \cdot 506) \times (10 \cdot 418) = 3030 \times 4180$ texture. However, for 61 disparities, no rectangular tiling exists, and a layout with 16×4 grids is allocated, of which the last three are not used (which wastes about 5 MB).

3.2.3. Regaining accuracy using a dichromatic approach

The dramatic speedup achieved by the DCB grid comes at some loss of quality. *Trading off speed* This is because the underlying bilateral grid only works on greyscale images and *for accuracy* hence does not differentiate colours that have similar greyscale values, as shown in the examples of Figure 3.4.

Colour discriminability can be increased by adding additional colour axes to the *Memory* grid. Unfortunately, the memory requirements of the bilateral grid are exponential *requirements* in the number of dimensions. The *Teddy* and *Cones* stereo images, for example, each have a total memory footprint of

$$60 \text{ disparities} \times \frac{450}{10} \times \frac{375}{10} \times \left(\frac{100}{10}\right)^{k} \times 8 \text{ bytes} \tag{3.9}$$

when using the standard parameters $\sigma_s = 10$ and $\sigma_r = 10$, k total colour dimensions, and two single-precision floating-point numbers per grid cell. For the greyscale DCB grid, where $k = 2$, this amounts to 78 MB. However, the best results, with full CIELAB colours in both images ($k = 6$), would require a prohibitive 764 GB.

Given this constraint, a total of at most $k = 3$ colour dimensions in both images *Dichromatic* can be afforded on current generation graphics cards, resulting in 783 MB for the *approach* *Teddy* stereo image. This allows one additional colour axis in one of the stereo half-images, in addition to each image's greyscale lightness component. The result is a *dichromatic* technique which can differentiate colours along two colour axes in one of the two images. This is an interesting trade-off between the common monochromatic and anthropocentric trichromatic stereo approaches, that has not previously been explored.
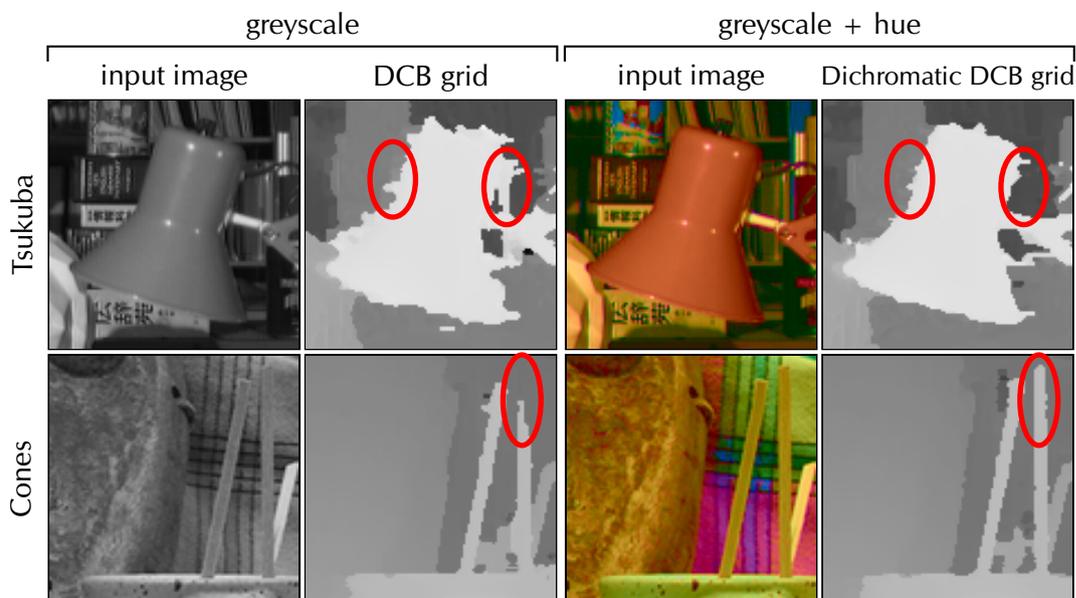
Figure 3.4: Comparison of the mono- and dichromatic DCB grid. The input images are displayed as 'seen' by the algorithms. Note that the disparity map of the dichromatic DCB grid visibly improves on the monochromatic DCB grid.

Comparison of colour dimensions A range of additional colour dimensions are evaluated in Table 3.1. The table compares the following seven candidate colour dimension: HSL hue and saturation, CIELAB chromaticities $a^\star$ and $b^\star$, and the derived properties hue h_{ab}, saturation s_{ab} and chroma $C_{ab}^\star$. The 'Rank' column shows the ranking each particular candidate would have achieved in the Middlebury stereo benchmark (Section 2.4.5). The best technique, in terms of lowest average rank, is CIELAB hue h_{ab}.

Principal components In follow-up work to mine, Zhu (2011) uses principal component analysis (PCA) to find the first two principal colour components of a stereo image instead of using the lightness $L^\star$ and hue h_{ab} as proposed in this chapter, which reportedly improves accuracy even further.

Technique	Rank	Tsukuba			Venus			Teddy			Cones		
		nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc
$h_{ab} = \mathrm{atan2}(b^\star, a^\star)$	**48.6**	**4.28**	5.44	14.1	1.20	1.80	9.69	9.52	16.4	19.5	4.05	**10.4**	**10.3**
HSL saturation	49.0	4.44	**5.37**	**12.9**	**1.05**	**1.58**	**8.29**	9.46	16.4	19.4	4.30	10.7	11.3
$C_{ab}^\star = \sqrt{a^{\star 2} + b^{\star 2}}$	49.9	4.97	5.94	16.7	1.15	1.75	8.65	9.55	16.4	19.9	**4.00**	**10.4**	10.5
$s_{ab} = C_{ab}^\star / L^\star$	50.0	4.36	5.45	12.9	1.19	1.86	9.32	**9.41**	**16.3**	**19.2**	4.41	10.8	11.6
$b^\star$	50.8	4.79	5.83	16.2	1.25	1.84	10.10	9.53	**16.3**	19.6	4.28	10.7	11.6
$a^\star$	52.0	5.36	6.49	18.3	1.24	1.84	9.13	9.62	16.5	19.9	4.28	10.5	11.3
HSL hue	51.2	4.62	5.85	14.9	1.30	1.87	10.40	9.83	16.6	20.2	4.18	10.7	11.1

Table 3.1: Accuracy comparison of the dichromatic DCB grid with various colour properties using the Middlebury stereo benchmark (Section 2.4.5), to 3 significant digits.

3.3. Still image results and applications

Introduction

This section considers the performance of the techniques presented so far, comparing them to other real-time stereo matching techniques in terms of run times and accuracy, and showing an application that benefits from the speed of the DCB grid. As in Yoon and Kweon's paper, the left-right post-processing is included when reporting accuracy figures, but not for the run time measurements.

Implementation

All techniques in this chapter are implemented using *C for CUDA* – NVIDIA's parallel computing architecture for general purpose computation on their GPUs[10]. The techniques are implemented against CUDA 2.3 and take advantage of newly introduced features such as atomic integer arithmetic in global memory for creating the DCB grids (see Section 3.2.2). Exact implementation details are omitted here, but the source code is publicly available on the project website[11].

Configuration

All results in this chapter were created using an NVIDIA Quadro FX 5800 graphics card with 4 GB video memory – the largest of any commercial GPU as of late 2009 (6 GB is the limit of 2012 graphics cards). The GPU was supported by a 2.4 GHz Intel Quad Core processor with 4 GB RAM.

3.3.1. Run time measurements

Relative speed

The run time measurements for the standard stereo datasets are shown in Table 3.2. My re-implementation of Yoon and Kweon's technique is about 25× faster than their reported figures, and 30 per cent faster than the full-kernel DCB aggregation. Relative to these techniques, the DCB grid is more than 165× and 200× faster, respectively. The DCB grid is also 14× faster than its dichromatic variant.

Absolute speed

The run times in Table 3.2 also show that the DCB grid runs at a frame rate of 13 Hz or higher on all datasets, with 70 Hz on the *Tsukuba* stereo image. This made the DCB grid the fastest stereo correspondence approach on the Middlebury evaluation website at the time of publication (September 2010).

Technique	Tsukuba 384×288×16	Venus 434×383×20	Teddy 450×375×60	Cones 450×375×60
DCB Grid	**14.2**	**25.7**	**75.8**	**75.0**
Real-time GPU (Wang et al., 2006)	30*	60*	200*	200*
Reliability DP (Gong and Yang, 2005)	42	109	300*	300*
Dichromatic DCB Grid	**188**	**354**	**1 070**	**1 070**
Plane-fit BP (Yang et al., 2008)	200*	400*	1 000*	1 000*
Y&K (my GPU implementation)	**2 350**	**4 480**	**13 700**	**13 700**
Full-kernel DCB	**2 990**	**5 630**	**17 700**	**17 600**
Yoon and Kweon (2006)	60 000	100 000*	300 000*	300 000*

Table 3.2: Run time comparison in milliseconds. Techniques implemented for this chapter are emboldened. Asterisks (*) mark run times estimated from reported figures, rounded to one significant digit.

---

[10] http://developer.nvidia.com/what-cuda

[11] http://richardt.name/dcbgrid/

3.3.2. Accuracy comparison

General   The disparity maps of all proposed techniques are shown in Figure 3.6 for visual comparison, and evaluated quantitatively on the Middlebury datasets in Table 3.3.

Full-kernel DCB aggregation   It is notable that the dual-cross-bilateral cost aggregation improves on my GPU implementation of Yoon and Kweon in the *nonocc* (non-occluded pixels) and *all* pixels categories in almost all cases. This is the highest ranked, but also slowest technique proposed in this chapter.

DCB grid   The dual-cross-bilateral grid is the lowest ranked technique amongst the real-time techniques in Table 3.3, but also the fastest. It performs particularly poorly on the *Tsukuba* image. But with its hand-labelled disparity map and low disparity range (16 levels), it is an unrealistic dataset. On the more realistic *Cones* image, with large disparity range (60 levels), the DCB grid performs reasonably competitively. The poor accuracy is caused by its operation in greyscale instead of full colour.

Dichromatic DCB grid   The dichromatic DCB grid improves on the monochromatic DCB grid in all categories, achieving results comparable (*Tsukuba*, *Teddy*) or superior (*Venus*) to the GPU implementation of Yoon and Kweon, at a 13× speedup. The close-ups in Figure 3.4 also show qualitative improvements. These results demonstrate that partial-colour solutions can improve stereo results, and I believe that this idea has more general applicability in computer vision.

Trade-off of run time versus accuracy   Tables 3.2 and 3.3 also show an interesting trade-off between a technique's run time and its accuracy (visualised in Figure 3.5): both 'Real-time GPU' (Wang et al., 2006) and 'Reliability DP' (Gong and Yang, 2005) are slower than the DCB grid, but faster than the dichromatic DCB grid, with accuracy being inversely related: the dichromatic DCB grid outperforms both 'Real-time GPU' and 'Reliability DP' which in turn outperform the DCB grid. Yang et al.'s plane-fit BP (2008) outperforms the dichromatic DCB grid at similar run times, but their technique occupies both CPU and GPU, whereas the proposed GPU-based techniques leave the CPU available for other tasks.

Technique	Rank	Tsukuba			Venus			Teddy			Cones		
		nonocc	all	disc	nonocc	all	disc	nonocc	all	disc	nonocc	all	disc
Plane-fit BP	19.4	0.97	1.83	5.26	0.17	0.51	1.71	6.65	12.10	14.7	4.17	10.70	10.60
Yoon and Kweon	32.8	1.38	1.85	6.90	0.71	1.19	6.13	7.88	13.30	18.6	3.97	9.79	8.26
Full-kernel DCB	47.7	3.96	4.75	12.90	1.36	2.02	10.40	9.10	15.90	18.4	3.34	9.60	8.26
Y&K (GPU impl.)	48.2	4.39	5.29	8.10	1.30	2.07	8.31	9.39	16.30	18.4	3.68	9.96	8.42
Dichr. DCB Grid	52.9	4.28	5.44	14.10	1.20	1.80	9.69	9.52	16.40	19.5	4.05	10.40	10.30
Real-time GPU	56.2	2.05	4.22	10.60	1.92	2.98	20.30	7.23	14.40	17.6	6.41	13.70	16.50
Reliability DP	59.7	1.36	3.39	7.25	2.35	3.48	12.20	9.82	16.90	19.5	12.90	19.90	19.70
DCB Grid	64.9	5.90	7.26	21.00	1.35	1.91	11.20	10.50	17.20	22.2	5.34	11.90	14.90

Table 3.3: Performance accuracy of the presented techniques to Yoon and Kweon (2006) and selected real-time techniques using the Middlebury stereo benchmark.
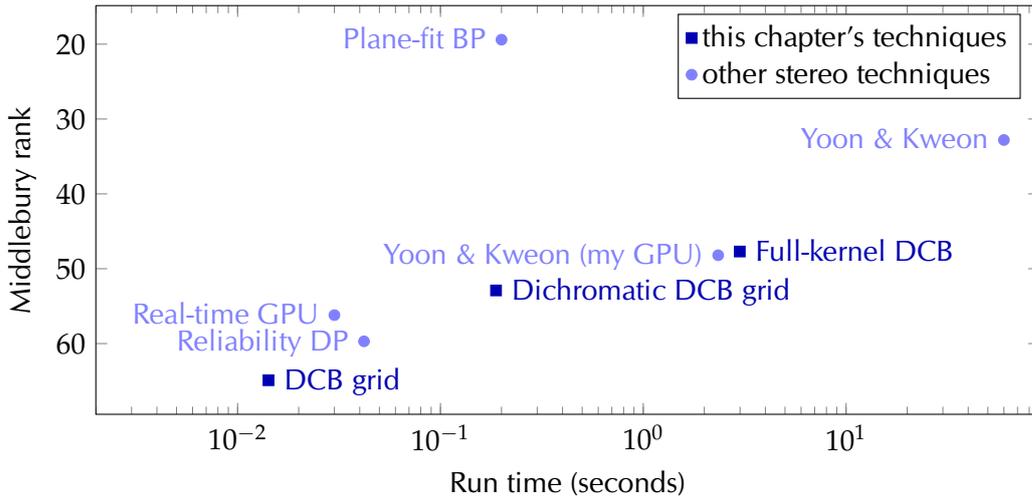
Figure 3.5: Scatter plot visualisation of run time (Table 3.2) versus Middlebury rank (Table 3.3) for the techniques presented in this chapter and other stereo matching techniques. Most techniques lie close to a straight line which trades off run time and Middlebury rank.
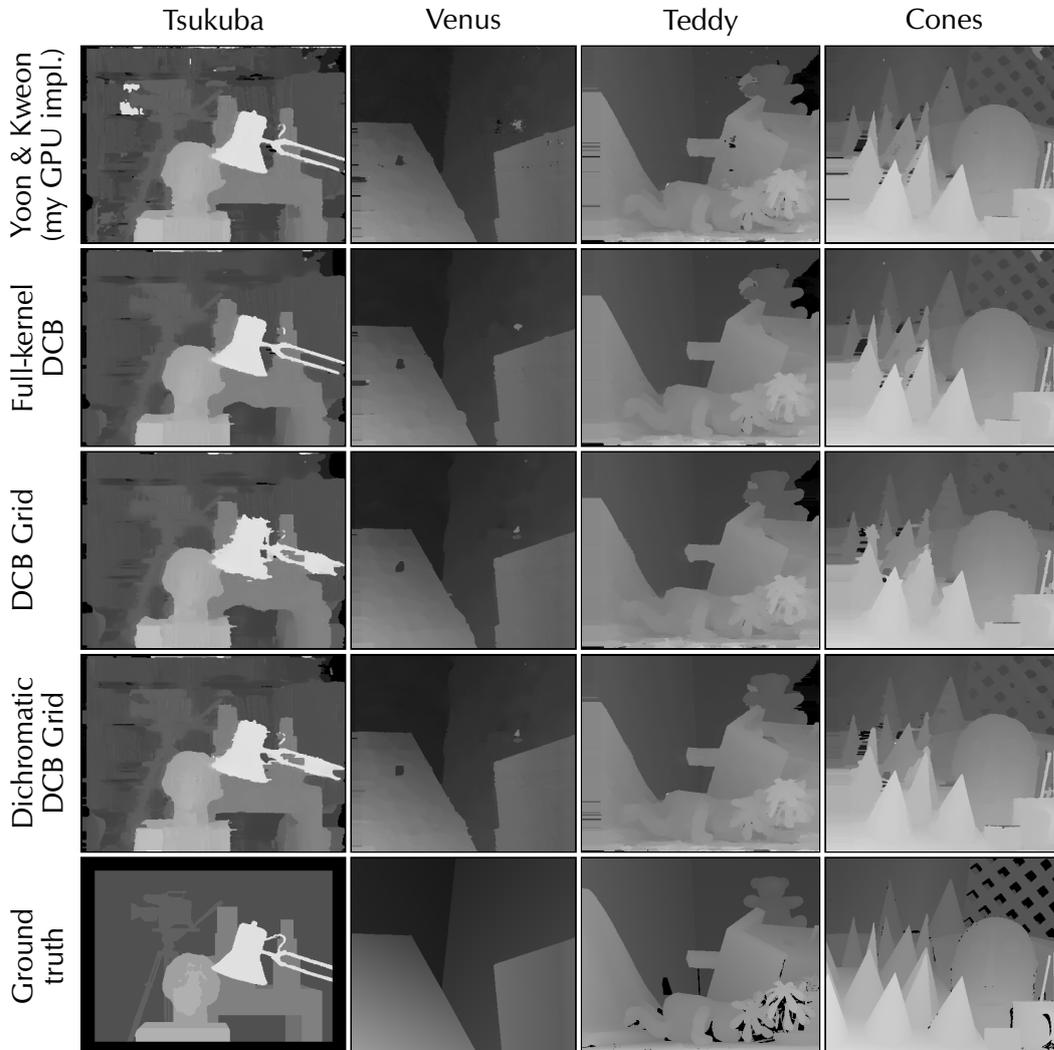
Figure 3.6: Disparity maps for the Middlebury datasets (Scharstein and Szeliski, 2002).

3.3.3. Application: spatial-depth super-resolution

Spatial-depth super-resolution Yang et al. (2007) use Yoon and Kweon's method as a central component in their spatial-depth super-resolution system. Starting from a low-resolution depth map, they iteratively upsample it to the full resolution of the input images using Yoon and Kweon's cost aggregation:

$D_0 \leftarrow$ up-sample disparity map using nearest neighbour interpolation
for iteration $i = 1$ to n **do**
    $C_i \leftarrow$ compute cost space from disparity map D_{i-1} (Equation 3.10)
    $C'_i \leftarrow$ aggregate costs based on C_i
    $D_i \leftarrow$ run sub-pixel winner-take-all on C'_i
end for

Cost space from disparity map At the start of each iteration, the cost space C_i is derived from the previous disparity map D_{i-1} using the truncated squared difference of disparities,

$$C(\mathbf{p}, d) = \min(\eta \cdot \Delta, (d - D(\mathbf{p}))^2), \qquad (3.10)$$

where $\eta = 0.5$ and Δ is the disparity range. The iterative refinement in the loop is typically executed $n = 3$ times.

Plug-in replacement for 100× speedup Modifying this algorithm to use the DCB grid instead results in a speedup of more than 100×. Figure 3.7 compares results, run times and errors.

Yoon and Kweon (2006)	My DCB Grid (Section 3.2.2)	Difference image

Hz	nonocc	all	disc		Hz	nonocc	all	disc
0.15	17.1	17.4	41.7		17.0	19.9	19.2	42.5
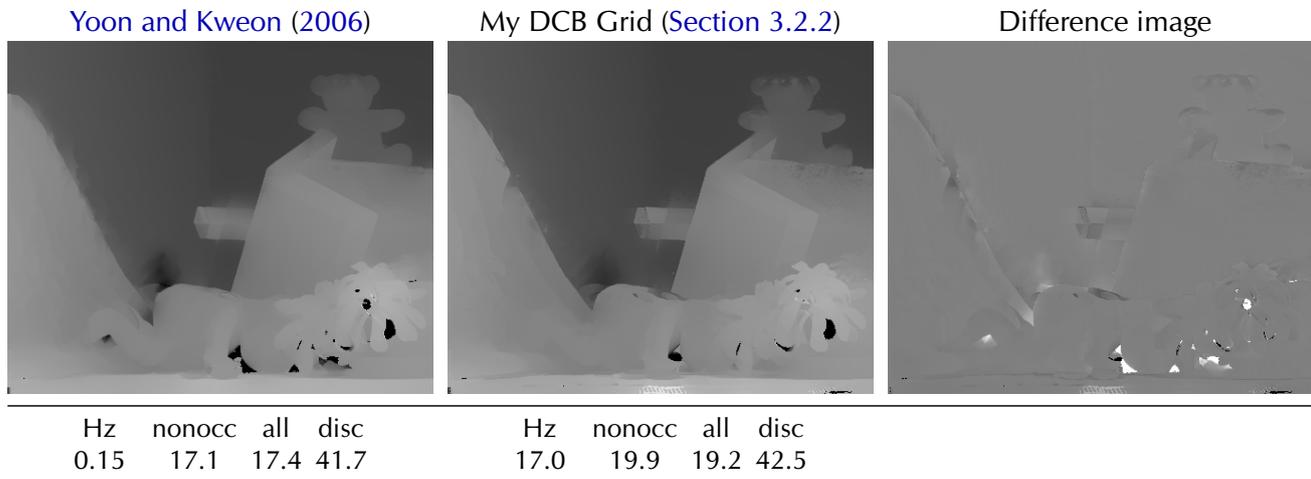
Figure 3.7: Comparison of cost aggregation techniques in Yang et al.'s spatial-depth super-resolution on 8× sub-sampled *Teddy*. My DCB grid is more than 100× faster, at only a small loss of quality.

3.4. Incorporating temporal evidence

Stereo videos pose different challenges to stereo images: the application of stereo matching techniques on a per-frame basis is generally insufficient to achieve flicker-free and temporally coherent disparity maps. This is because variations in the two videos over time, such as noise, may result in multiple disparity hypotheses with identical costs. Which one of these disparities 'wins' a pixel can change over time, and hence cause flickering.

Difficulties of stereo videos

Given the speed and success of the DCB grid method, this section turns its attention to introducing time as an additional dimension to the DCB grid. This approach is inspired by 'spacetime stereo' algorithms (Davis et al., 2005; Zhang et al., 2004) that aggregate costs over a 3D spatiotemporal support window, instead of just a spatial window in the current video frame.

Temporal DCB grid

For each frame of the video, the DCB grid is created and processed as in Section 3.2.2. However, the slicing stage linearly combines costs from grids of several frames, each weighted by w_i:

Formulation

$$C'(\mathbf{p}, d) = \sum_i w_i \cdot \Gamma_{d,i}\left(\frac{x}{\sigma_s}, \frac{y}{\sigma_s}, \frac{L_L^\star(\mathbf{p})}{\sigma_r}, \frac{L_R^\star(\overline{\mathbf{p}})}{\sigma_r}\right). \tag{3.11}$$

The following assumes a *streaming* approach to video processing, in which new video frames become available as soon as they are decoded from a file or recorded by a camera. In this approach, any upcoming frames are not available, as they would be from the 'future'. This is also known as *causal* video processing, as the current frame can only depend on frames which came before it.

Streaming approach

Let the temporal DCB grid in Equation 3.11 sum over $i \in [1-n, 0]$, where $i = 0$ stands for the current frame, $i = -1$ the previous frame and so on. Empirically, a window of $n = 5$ frames works well for videos with a frame rate of 30 frames per second. Each grid $\Gamma_{d,i}$ is sliced at the same coordinate $(x/\sigma_s, y/\sigma_s, L_L^\star(\mathbf{p})/\sigma_r, L_R^\star(\overline{\mathbf{p}})/\sigma_r)$, to extract the aggregated costs of a disparity hypothesis d at a pixel $\mathbf{p} = (x, y)$, but for a particular frame $i \in [1-n, 0]$. Each of these costs is then weighted by a factor w_i.

Equation 3.11 explained

The original spacetime stereo approaches (Davis et al., 2005; Zhang et al., 2004) use constant weights ($w_i = 1$) for all frames. But Gaussian weights, $w_i = \exp(-i^2/2\sigma_t^2)$ with $\sigma_t = 2$, work better and also extend the DCB grid into the time dimension. I also tried Paris' adaptive exponential decay (2008), but did not see improvements.

Temporal weights w_i

There are several practical limitations of this approach. Firstly, the dichromatic and temporal extensions of the DCB grid cannot be used at the same time, as there is insufficient memory to handle six dimensions of data. Secondly, the spatiotemporal support does not compensate for object motion. And lastly, the key assumption of the temporal DCB grid extension is that pixels with similar colours have similar disparities, both across space and time. Although this assumption is valid in many cases, if it is violated, results may suffer.
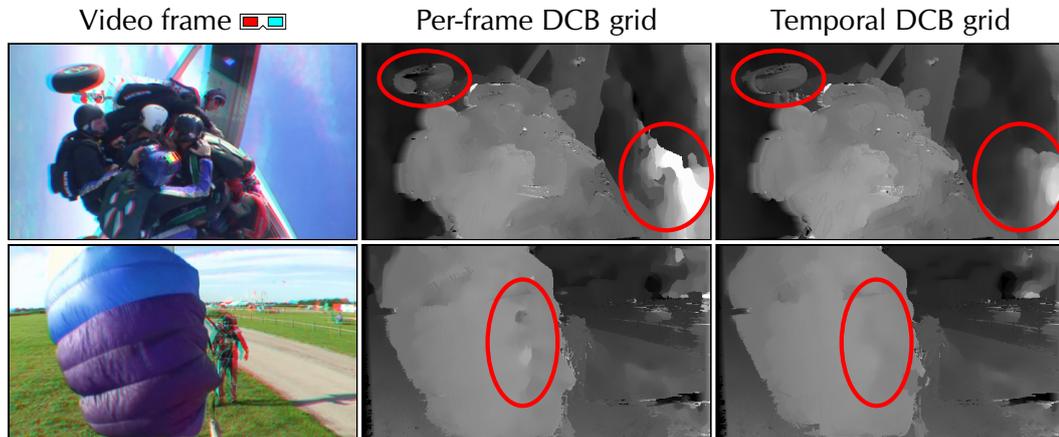
Limitations

Results of qualitative and quantitative nature are discussed next. The temporal DCB grid is evaluated qualitatively using real stereo videos and quantitatively on synthetic stereo videos with ground truth disparities, where it is also compared against per-frame techniques.

Evaluation

3.4.1. Qualitative Evaluation

Skydiving video Figure 3.8 shows frames from a skydiving video, processed at a spatial resolution of 480×270 with 40 disparities and without the left-right consistency check. On the test machine (Section 3.3), the per-frame DCB grid runs at 16 Hz and the temporal DCB grid at 14 Hz. As can be seen in the paper's supplementary videos[12], the temporal DCB grid visibly reduces flickering compared to the per-frame method.

Video frame ▣▣	Per-frame DCB grid	Temporal DCB grid

Figure 3.8: Disparity maps for selected frames of the 'skydiving' stereo video. Note that the temporal DCB grid visibly reduces errors (see highlighted regions). Video © Eric Deren, Dzignlight Studios.

3.4.2. Quantitative Evaluation

Synthetic stereo videos The quantitative evaluation of disparity maps from stereo videos is hindered by the general lack of ground truth disparity maps. For this reason, Ian Davies and I created a set of five stereo videos with ground truth disparity maps (see Figure 3.9).

Video design We generated the sequences using Blender, an open source modeller. Each frame is 400×300 pixels in size with a disparity range of 64 pixels. The *Book*, *Tanks* and *Temple* objects were taken from the Official Blender Model Repository[13], while the *Tunnel* scene was our own design. For the *Street* sequence, we combined models and materials by Andrew Kator and Jennifer Legaz[14]. We added two parallel cameras to each scene with a small lateral offset between them, to provide the left and right views, and used the Blender node system to render disparity maps from the point of view of each camera.

Rendering disparity Internally, Blender uses a z-buffer and thus only works with depth, not disparity. I determined the mapping from depth z to disparity d to be $d = (w \cdot f \cdot b)/(s \cdot z)$, where w is the frame width in pixels, f the focal length (in mm), b the baseline of the two cameras (in Blender units) and $s = 32$ mm the sensor size. I found this experimentally by extracting corresponding depth extrema in each scan-line of the two images, whose distance along the scan-line is the disparity equivalent to their depth.

---

[12] http://richardt.name/dcbgrid/supplement/

[13] http://e2-productions.com/repository/

[14] Licensed under CC-BY 3.0, available at http://www.katorlegaz.com/3d_models/.

video frame   disparity map

Book (41 frames)

In this sequence, a page of an old book is turned, which tests how deformable objects are handled.

Street (100 frames)

This sequence pans across a street view, testing translations 'in the camera plane'.

Tanks (100 frames)

Here, the camera flies along a grid of tanks, testing translations 'out of the camera plane'.

Temple (100 frames)

This sequence shows a rotating Mayan temple, which is testing rotational motions.

Tunnel (100 frames)

In this sequence, the camera moves through a tunnel, which tests zooming camera motions.

Figure 3.9: Overview of the synthetic stereo videos with ground truth disparity maps (left views).

Using the synthetic ground truth videos, the temporal DCB grid can be compared quantitatively against per-frame techniques. All videos were processed using all techniques, including the same left-right consistency post-processing as earlier.   *Evaluation setup*

The ground truth stereo videos are noise-free, but real videos are not. Therefore, the robustness of per-frame techniques and the temporal DCB grid to noise was analysed first. To simulate thermal imaging noise, zero-centred Gaussian noise was added to all colour channels of the input frames. The accuracy and run times of all implementations are shown in Table 3.4. The level and variability of errors is summarised using the mean and standard deviation of the percentage of bad pixels across frames.   *Performance on noisy videos*

Technique	Time	*Book*		*Street*		*Tanks*		*Temple*		*Tunnel*	
	in ms	mean	stdev	mean	stdev	mean	stdev	mean	stdev	mean	stdev
Temporal DCB Grid	90	**44.0**	2.02	**25.9**	**2.00**	**31.4**	6.06	**31.7**	1.82	36.4	**7.88**
DCB Grid	51	52.2	2.04	32.5	2.33	36.0	6.16	39.5	1.91	**25.7**	11.10
Dichromatic DCB Grid	782	58.9	1.83	39.2	2.62	47.8	12.00	43.0	1.73	32.9	12.00
Full-kernel DCB	13 200	65.9	1.45	49.1	3.13	53.5	6.15	52.0	**1.28**	43.0	11.70
Y&K (my impl.)	9 770	84.2	**1.24**	56.1	2.67	87.7	**2.01**	72.8	1.80	58.4	11.70

Table 3.4: Accuracy comparison of the proposed methods with additive Gaussian noise ($\sigma = 20$). Shown are the average and standard deviation of the percentage of bad pixels (threshold is 1), and per-frame run times. For most datasets, the temporal DCB grid has the lowest mean error.

Ranking of techniques The best results are produced by the temporal DCB grid which significantly outperforms the per-frame techniques on all datasets except 'tunnel', on which it shows the least variation in error. The per-frame DCB grid techniques come second and third, and the full-kernel implementations are placed last.

Analysis of accuracy The relatively poor accuracy of the temporal DCB grid on the 'tunnel' video is likely because it has a lot of texture, so that simple per-frame approaches work well, while the temporal DCB grid tends to over-smooth. The camera motion also violates the assumption that similar colours correspond to similar disparities. Nevertheless, it reduces flickering visibly in all videos, as can be seen in the supplementary videos[15].

Sub-linear run time It is also notable that the temporal DCB grid has a run time that is sub-linear in the number of frames: it only takes 76 per cent longer than the per-frame DCB grid to process a five frame window instead of a single frame.

Better accuracy on noisy videos Plots of the error levels at noise standard deviations between 0 and 100 (out of 255) are shown in Figure 3.10. The graphs show that the temporal DCB grid improves on the per-frame technique at increased noise levels in all cases. In particular, it is superior for all noise levels in the 'street' and 'temple' sequences, and starting from noise levels of 5–45 for the other sequences. It is the integration of temporal evidence across several frames that makes this improvement possible.
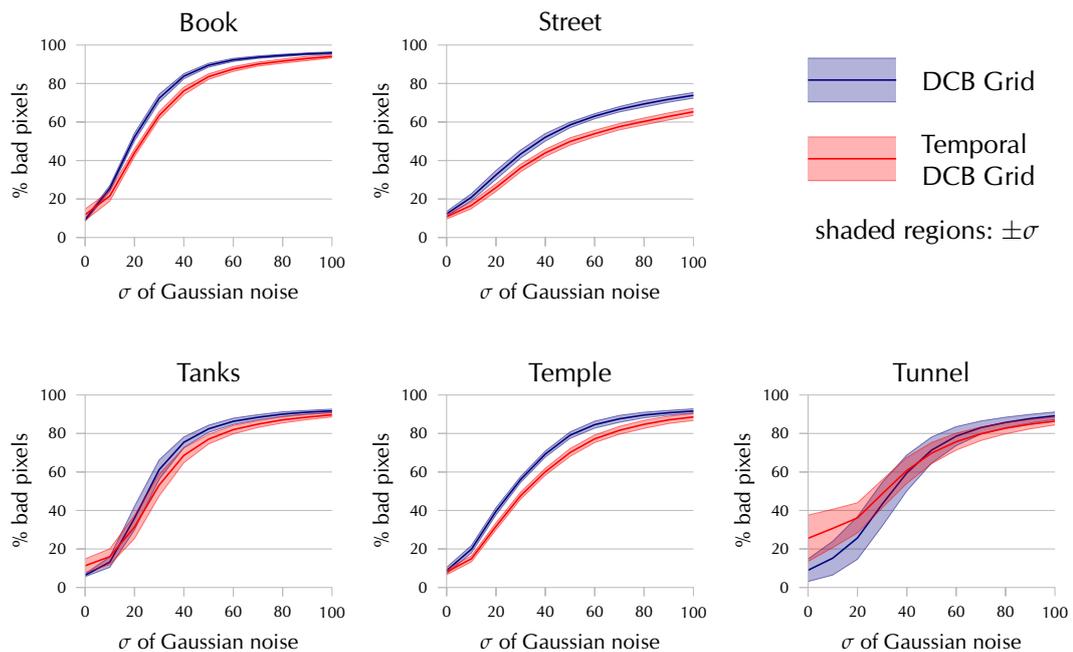
Figure 3.10: Error versus noise curves for ground truth stereo videos: the temporal outperforms the per-frame DCB grid at higher noise levels. See the supplementary video[15] for a visual comparison.

---

[15] http://richardt.name/dcbgrid/supplement/

3.5. Conclusion

Rewriting Yoon and Kweon's adaptive support weights as a dual-cross-bilateral filter with Gaussian weights enables the use of the bilateral grid for acceleration. The DCB grid achieves real-time frame-rates through a speedup of more than $200\times$ compared to a full-kernel GPU implementation, at only a small loss of precision.

Reformulation for acceleration

The speed of the DCB grid makes it versatile. Techniques building on Yoon and Kweon's method automatically benefit from a large speedup. I showed this by applying it to Yang et al.'s spatial-depth super-resolution, achieving a speedup of $100\times$, with minimal loss of quality.

Faster applications

The DCB grid also extends into the temporal domain, aggregating support over adaptive spacetime support windows. It outperforms per-frame techniques in the presence of noise in the input images, and enforces temporal coherence under the assumption that pixels of similar colour in consecutive frames have a similar depth.

Spatiotemporal stereo matching

To evaluate stereo matching on videos, we introduced five computer-generated stereo videos with ground truth disparity maps. We hope that these videos will enable others to develop and evaluate new and improved techniques that are aimed at solving the challenges of stereo videos. In fact, some researchers already use our videos for evaluation (Khoshabeh et al., 2011; Hosni et al., 2011).

Evaluating stereo matching on videos

The source code for all techniques, the ground truth stereo videos and further supplementary materials are available from the project website[16].

Published code & data

The dichromatic DCB grid showed that colour is a useful component in achieving high quality disparity maps. However, the enormous memory requirements of the bilateral grid preclude filtering in full colour. Recent work by Adams et al. (2010) proposes a method with linear memory requirements. They confirm that the bilateral grid is the fastest technique for 4D bilateral filtering with a standard deviation of 10, as used by the DCB grid. However, full-colour 8D filtering would be about $4\times$ faster with their technique, with significantly reduced memory requirements.

Future work: full-colour filtering

We hope that our new ground truth stereo videos provide a useful resource for research in depth estimation from stereo videos. There is a need for specialised stereo video correspondence techniques that incorporate temporal evidence to resolve ambiguities. With this in mind, it will be necessary to set up a stereo video evaluation website, perhaps as part of the Middlebury vision website. For this, one also needs to find metrics that objectively quantify flickering and temporal coherence in disparity videos.

Future work: Evaluation website for stereo videos

Despite the increased temporal coherence offered by the technique developed in this chapter, it became apparent that the quality of the disparity map is insufficient for more demanding applications such as some video effects described in Chapter 5. For this reason, I take a step back in the next chapter and investigate how to use depth data from a time-of-flight camera (Section 2.3) to overcome the shortcomings of approaches based on stereo matching.

Next chapter

---

[16] http://richardt.name/dcbgrid/

Coherent depth from time-of-flight cameras

<div style="float:right">**4**</div>

This chapter presents research that was carried out during a research visit at the Max-Planck-Institut Informatik in Saarbrücken, Germany, in collaboration with Carsten Stoll and Christian Theobalt.

The work has been accepted at Eurographics 2012 and will be published in a special issue of Computer Graphics Forum (Richardt et al., 2012).

Time-of-flight cameras acquire distance maps by timing how long light takes to travel from the camera into the world and back. This approach makes the quality of distance maps largely independent of textures in the scene – as long as objects reflect light in the relevant range of wavelengths. In this respect, time-of-flight cameras improve on stereo matching techniques, as stereo techniques often have difficulties in weakly or periodically textured regions. However, the main disadvantage of time-of-flight cameras is their low spatial resolution (such as 176×144) and high noise levels (± 1 cm). *Introduction*

The principal idea in this chapter is to combine a time-of-flight camera with a synchronised high-resolution video camera and to merge both video streams into a coherent RGBZ video – a video with plausible per-pixel depth at colour video resolution, with strongly reduced noise level and correspondence over time. This requires aligning the two input streams, upsampling and denoising of the depth stream, and making the depth stream coherent over time. *Motivation*

Diebel and Thrun (2006) were among the first to fuse data from a low-resolution range scanner and a high-resolution colour camera. They infer an upsampled and denoised depth map using Markov Random Fields by observing that strong colour and depth edges often coincide. This assumption is also exploited by more recent approaches based on the bilateral filter (Section 2.5). Joint-bilateral upsampling (Kopf et al., 2007) is one such depth super-resolution approach which evaluates the data and range terms on depth and intensity channels respectively. Another approach is Yang et al.'s spatial-depth super-resolution (2007), which was discussed in Section 3.3.3. All these techniques have run times of several seconds per frame. *Related work: old & slow*

Related work: new & fast

Chan et al. (2008) proposed a real-time technique for joint-bilateral filtering that locally adjusts the filter to simple noise estimates. All these techniques only consider single frames from a video, and do not exploit the temporal coherence of video streams to further reduce noise levels, as is done in this chapter. Dolson et al. (2010) upsample sparse data acquired from a laser range finder using a spatio-temporal joint-bilateral filter for interpolating missing data. In contrast, the filtering approach in this chapter performs spatiotemporal filtering and super-resolution on time-of-flight distance maps, which are dense, but much noisier.

Challenges

This chapter presents a novel fast geometry filtering approach that fuses the colour and depth videos captured from a prototype camera (Section 4.1) into a coherent RGBZ video. Note that the aim is not geometric accuracy, but plausible geometry which is sufficient for many of my applications, such as non-photorealistic rendering. The four main challenges to address are:

– **Video alignment**
  The videos are captured from laterally displaced viewpoints, and thus capture different views of a scene. To combine the videos, they first need to be aligned.

– **Half-occlusions**
  Different viewpoints also mean that each camera can see into areas which are occluded in the other camera's view. This is illustrated in Figure 4.2 (right).

– **Resolution mismatch**
  The depth data captured by the time-of-flight camera has a spatial resolution of 176×144, whereas the video camera has a maximum resolution of 1024×768.

– **Noisy depth data**
  The accuracy of the depth data is ±1 cm, but this is masked by extreme temporal fluctuations with a standard deviation of several centimetres (see Figure 4.8).

Physical limitations

The limitations of time-of-flight cameras are caused by the technical difficulty of measuring the short time intervals associated with light travelling only a few metres through air. In vacuum, light travels at a speed of around 30 cm/ns, so to achieve better distance precision, the camera sensor needs to read out pixels more quickly. The number of photons per time interval then limits the pixel size on the sensor, as reducing the pixel size would decrease the signal-to-noise ratio.

Structure of this chapter

This chapter presents a video processing pipeline (illustrated by Figure 4.1) that addresses the four challenges above. The colour and depth videos are first aligned using a rigid transform (Section 4.1), half-occluded areas are invalidated and filled in again (Section 4.2), and a novel spatiotemporal filter performs super-resolution and denoising simultaneously (Section 4.3).
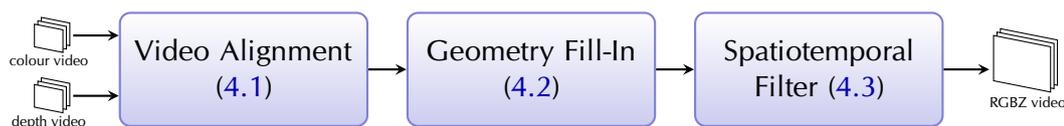

Figure 4.1: The RGBZ video processing pipeline that combines a high-resolution colour video with a noisy, low-resolution depth video into a coherent RGBZ video.

4.1. Aligning the colour and depth videos

I built a prototype camera at MPI Informatik which comprises a MESA Imaging SR4000 time-of-flight camera and a Point Grey Flea2 colour camera (Figure 4.2, left). The cameras are fitted side by side to minimise their baseline, their optical axes are aligned to be almost parallel to achieve similar fields of view, and I adjusted the video camera's lens to cover the time-of-flight camera's fixed field of view.

Prototype camera
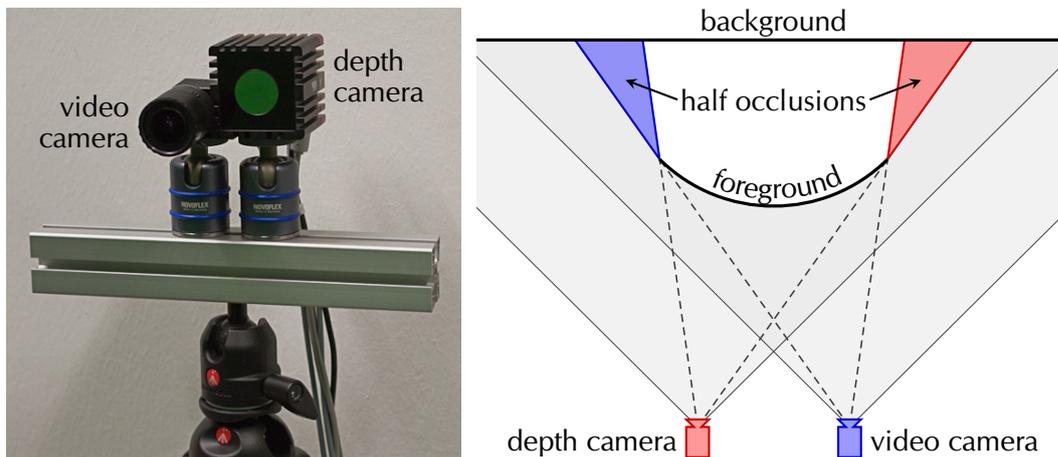
Figure 4.2: Left: The prototype camera setup with video and depth cameras mounted on a tripod.
Right: Schematic of the camera setup, with highlighted half-occlusion areas.

The prototype camera allows full hardware and software control over all modules. The two cameras are synchronised using a custom circuit which connects the video camera's strobe output to the time-of-flight camera's trigger input. Due to limitations of the trigger circuitry, the setup is limited to capture video at 15 Hz, which could be overcome with additional engineering. However, this frame rate is sufficient for demonstrating many interesting RGBZ video processing tasks.

Synchronisation

Other sensors are now commercially available, such as the Microsoft Kinect[17] which is the first mass-market product to combine an IR-based active stereo system with a colour video camera. Both the prototype camera and the Kinect suffer from the same general noise problem – time-of-flight depth data are contaminated by measurement noise whereas Kinect data are quantised in depth. Both cameras also use separate depth and video cameras. Therefore, the fill-in and filtering techniques in Sections 4.2 and 4.3 are needed in the same manner. While I explain these techniques using the prototype camera, the very same approach is also applicable to the Microsoft Kinect, as shown in Section 4.4.

Microsoft Kinect

The alignment of the colour and depth videos involves the choice of a common reference frame. Using the view of one of the two cameras limits occluded areas (see Figure 4.2, right) to be of a single type: either colour or geometry would be occluded in the reference view; any other reference frame would result in a mix of these occlusions, which would be harder to deal with than a single type. This leaves

Reference frame

---

[17] The development of the prototype camera system and the filtering techniques presented in this chapter precede the November 2010 launch of the Microsoft Kinect by a few months.

two options: (1) aligning the depth video to the colour video using reprojection; or (2) aligning the colour video to the depth video using projective texturing as in Lindner et al. (2007). I chose the first option, as the goal is to upsample and denoise the distance map using the more reliable image data. Projective texturing would reduce the quality of the available colour image and hence also the filtered distance map, due to texture interpolation and unavoidable texturing artefacts.

Camera calibration   To reproject the distance map to the video camera's view, the cameras were first calibrated using standard tools[18]. Video frames from both cameras are undistorted using the intrinsic camera calibration parameters on the fly. Time-of-flight cameras also exhibit a systematic depth bias and more sophisticated calibration approaches exist to address this (Kolb et al., 2010, section 4). However, we found the basic calibration to be sufficient, because of the small baseline between the colour and depth cameras which makes the influence of systematic depth errors negligible.

Notation: depth versus distance   The terms 'depth' and 'distance' are often used interchangeably, so let me again clarify what is meant by each. The term 'distance' (or 'range') denotes the Euclidean distance between a 3D world point and the camera centre, whereas 'depth' refers only to the distance along the viewing direction. Points at constant distance form a sphere, and points at constant depth a plane. Depth maps can be converted to and from distance maps if the intrinsic camera calibration is known.

Alignment by reprojection   Using the calibrated time-of-flight camera parameters, the distance map is back-projected to the world coordinate frame as a triangle mesh that connects the centres of neighbouring pixels in the distance map like a 'rubber sheet' (▦). This triangle mesh, in world-space coordinates, is then projected into the video camera's view.

Reprojection in OpenGL   Specifically, a vertex buffer of the size of the distance map is first created, with simple triangle connectivity described by the corresponding index buffer. For each new video frame, the vertex shader uses the distance map to position the vertices in the vertex buffer as specified in the distance map. The OpenGL model-view matrix is set to the relative transform from the time-of-flight to the video camera, as given by the extrinsic camera parameters, and the projection matrix is defined by the colour video camera's intrinsic camera parameters.

Distance map format   The output is an aligned distance map at colour video resolution, but the true visible detail has not been increased, just linearly upsampled. It is stored as an RGBA32 texture in OpenGL, where the first three components encode the 3D coordinates and the fourth the distance of a pixel. If a pixel's distance is set to zero, it is invalid. In this case, the 3D coordinates of the pixel encode the viewing ray through the pixel, so that the pixel may be filled in in the correct location in the next stage.

---

[18] Initially using Jean-Yves Bouguet's Matlab Camera Calibration Toolbox, later using OpenCV.

4.2. Filling in invalid geometry

The time-of-flight and video cameras capture slightly different views due to their displacement. This results in regions occluded in one camera's view that are visible in the other view (as in Figure 4.2, left). Projecting the distance map onto the video camera's view thus introduces holes that need to be filled. *Introduction*

The reprojection of the distance map in the previous section is performed using a 'rubber sheet' triangle mesh. In regions that are occluded in the time-of-flight camera's view, this geometry slopes to the background instead of showing a clear depth discontinuity (see Figure 4.3a). The time-of-flight camera also introduces so-called *flying pixels* at depth discontinuities which fluctuate at intermediate distances (Kolb et al., 2010) and need to be removed. The first step is hence to invalidate half-occluded and unreliable regions (Figure 4.3b), and then fill them in again from the surrounding geometry with the help of the colour image (Figure 4.3c). *Motivation & outline*
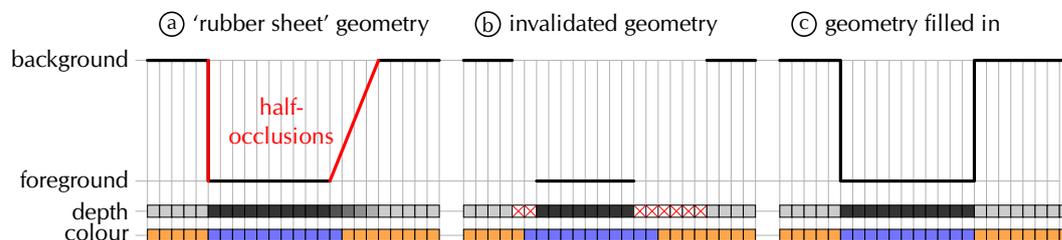
Figure 4.3: Illustration of the geometry fill-in procedure on a slice of scene geometry as seen from above: (a) 'rubber sheet' with half-occlusions; (b) after geometry invalidation; (c) with geometry fill-in.

Half-occlusions could be prevented by using co-linear optics for depth and colour video capture, for instance using a beam splitter, but flying pixels would still occur. Furthermore, all current systems, including the Kinect, use two displaced cameras. An algorithmic solution to overcome the resulting half-occlusions is thus highly relevant and ensures applicability to a wider range of possible RGBZ camera setups. *Co-linear optics*
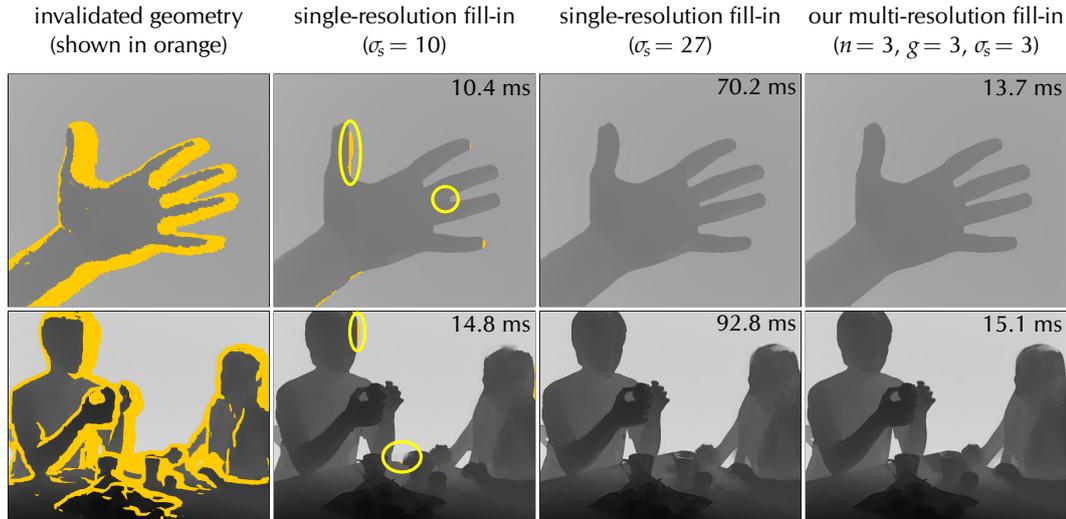
Pixels near depth edges in the time-of-flight camera's view are generally inaccurate as these *flying pixels* have a depth value somewhere between the front and the back surfaces covered by the pixel (Kolb et al., 2010). These pixels are thus removed by thresholding the gradient magnitude of the depth map, as approximated by a 3×3 Sobel filter with a threshold in $[0.1, 0.2]$, assuming distances in metres. In some cases, like the 'hand' sequence, visibly too little geometry is invalidated. In that case, I instead choose to use the surface normal to discard pixels which diverge too much from the view direction, as these tend to be noisy in time-of-flight cameras. *Geometry invalidation*

The next step is to fill the holes in the aligned distance map. The key assumption is again that depth and colour discontinuities coincide – a hypothesis exploited by joint-bilateral filters (Section 2.5.2). To fill large holes, as in the distance maps, a large filter radius ($\sigma_s > 25$) is needed, which precludes fast online processing. Instead, this section proposes a new multi-resolution joint-bilateral fill-in algorithm which is inspired by joint-bilateral upsampling (Kopf et al., 2007), and produces results of comparable quality but is suitable for online processing (see Figure 4.4). *Multi-resolution geometry fill-in*

invalidated geometry (shown in orange)	single-resolution fill-in ($\sigma_s = 10$)	single-resolution fill-in ($\sigma_s = 27$)	our multi-resolution fill-in ($n = 3$, $g = 3$, $\sigma_s = 3$)
	10.4 ms	70.2 ms	13.7 ms
	14.8 ms	92.8 ms	15.1 ms

Figure 4.4: The multi-resolution geometry fill-in technique is $5-6\times$ faster than the single-resolution fill-in with the large kernel size ($\sigma_s = 27$) necessary to fill holes, with comparable quality. A smaller kernel ($\sigma_s = 10$) has a similar run time, but shows errors (highlighted in yellow).

Algorithmic description The multi-resolution fill-in uses n resolution levels: 0 to $n-1$ from fine to coarse. To aid with the following explanation, a concrete example using $n = 3$ resolution levels in shown in Figure 4.5. Each level k has two inputs and one output, all of the same spatial resolution: the colour image i_k and aligned distance map d_k are inputs, and the filled-in distance map f_k is the output. The coarsest level, $n-1$, fills invalid pixels in the distance map d_{n-1} based on the corresponding colour image i_{n-1} using a standard joint-bilateral filter, resulting in the filled-in distance map f_{n-1}. All levels except the coarsest one, that is $k = 0, \ldots, n-2$, work as follows:

Downsampling & recursive call ① The image i_k and distance map d_k are downsampled by using every g^{th} pixel along the x and y axes, resulting in the downsampled versions i_{k+1} and d_{k+1}. These are passed to the next lower level, $k+1$, which returns a filled-in distance map f_{k+1} after all recursive processing has finished.

'Sparse' upsampling ② The coarser levels have recursively filled all invalid pixels of d_{k+1} in f_{k+1}. These newly-filled pixels are now upsampled to a sparse grid of pixels, which is used to fill invalid regions in the distance map d_k. This results in the 'sparsely upsampled' distance map u_k with filled-in values at every g^{th} invalid pixel.

Fill in pixels ③ The same joint-bilateral filter as at the coarsest level is applied to the sparsely upsampled distance map u_k and the image i_k to fill in all invalid pixels in d_k. Note that this recomputes the sparsely upsampled pixels to avoid artefacts.

Used parameters The results shown in this chapter use filter parameters $\sigma_s = 10$ and $\sigma_r = 0.05$. Most sequences use $n = 3$ levels, with resolution halving at each level ($g = 2$), except for the 'hand' sequence which has the largest half-occlusion regions and uses $n = 4$ resolution levels with threefold downsampling ($g = 3$), just like Figure 4.5 but with one additional resolution level. This is necessary to ensure good results across all video frames. Additionally, to reduce the influence of noise in the colour image, it is first filtered bilaterally using parameters $\sigma_s = 3$ and $\sigma_r = 0.1$.
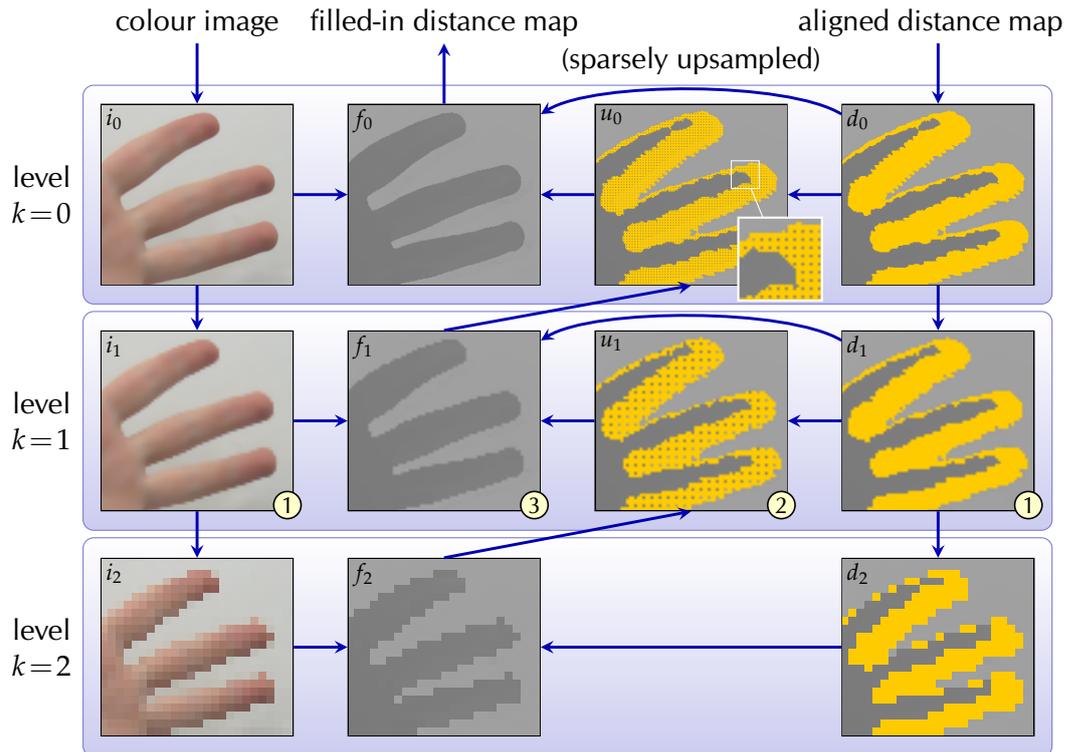
Figure 4.5: Illustration of the multi-resolution geometry fill-in technique using the 'hand' sequence (with parameters $n = g = \sigma_s = 3$). Please refer to Section 4.2 for a step-by-step explanation.

The approach described in this section uses a multi-resolution joint-bilateral filter to fill in invalidated regions in the aligned distance map. A potential alternative to this interpolation approach are inpainting algorithms (Bertalmio et al., 2000; Criminisi et al., 2004). These could extend background surfaces in invalidated regions by matching the slant of surrounding surfaces, which this section's approach cannot.

Alternative approaches

4.3. Spatiotemporal geometry filtering

Introduction The result of the geometry fill-in step is passed into a spatiotemporal filter which simultaneously denoises and super-resolves distance maps: the denoising step strongly reduces the spatial and temporal noise which is contained in distance maps recorded by time-of-flight cameras; and the super-resolution step uses the high-resolution colour video to increase the spatial resolution of distance maps by exploiting the coincidence of colour and depth edges.

Motivation A standard joint-bilateral filter applied at a single time step reduces spatial noise while preserving image and depth edges, but flickering caused by time-independent noise persists (Figure 4.8). To overcome the temporal noise, a spatiotemporal filter is proposed in this section, which incorporates information from previous frames.

Motion compensation Videos may contain significant motion and the filter therefore needs to be motion-compensated. A related technique in this context is Herzog et al.'s spatiotemporal upsampling technique (2010) for efficient high-resolution rendering. Their technique uses a single motion-compensated sample from the previous filtered frame to filter the current frame. While this works well on the clean rendered geometry, it performs poorly on the noisy distance maps acquired from time-of-flight cameras. Better results are achieved using the filter described in this section, which motion-compensates all kernel pixels, not just the centre, as illustrated in Figure 4.6.

Structure & notation In the following, I will first explain a purely spatial version of the filter as a baseline, and then extend it to the temporal domain. The following uses the homogeneous notation of the bilateral filter described in Section 2.5.1, where values are represented as homogeneous quantities and the homogeneous coordinate is filtered like the others, as this eliminates the division by the sum of weights in the filter notation.

4.3.1. The spatial joint-bilateral geometry filter

Spatial filter definition The distance map can be filtered in one time step using a dual-joint-bilateral filter, which preserves edges in the colour image $\mathbf{i}(\mathbf{p}, t)$ and the distance map $d(\mathbf{p}, t)$. This is conceptually similar to the dual-cross-bilateral filter described in Section 3.1.2. The spatially filtered distance for pixel $\mathbf{p}$ at time step t is given by

$$f_s(\mathbf{p}, t) = \sum_{\mathbf{q} \in N_\mathbf{p}} w_c(\mathbf{p}, \mathbf{q}) \cdot w_d(\mathbf{p}, \mathbf{q}) \cdot w_s(\mathbf{p}, \mathbf{q}) \cdot d(\mathbf{q}, t), \qquad (4.1)$$

where $N_\mathbf{p}$ is the set of pixels in the kernel of radius $2\sigma_s$ centred on $\mathbf{p}$, and the colour, distance and spatial weights are given by

$$w_c(\mathbf{p}, \mathbf{q}) = G_{\sigma_c}(\|\mathbf{i}(\mathbf{p}, t) - \mathbf{i}(\mathbf{q}, t)\|), \qquad (4.2)$$

$$w_d(\mathbf{p}, \mathbf{q}) = G_{\sigma_d}(|d(\mathbf{p}, t) - d(\mathbf{q}, t)|), \qquad (4.3)$$

$$\text{and} \quad w_s(\mathbf{p}, \mathbf{q}) = G_{\sigma_s}(\|\mathbf{p} - \mathbf{q}\|), \qquad (4.4)$$

where, as before, $G_\sigma(x) = e^{-x^2/2\sigma^2}$.

Pixel indices The pixels indices $\mathbf{p}$ and $\mathbf{q}$ are always in the current frame, at time t, whereas $\overline{\mathbf{p}}$ and $\overline{\mathbf{q}}$ (which are introduced next) are in the previous frame $(t-1)$. Typical filter parameters are $\sigma_c \in [0.05, 0.1]$, $\sigma_d \in [0.075\,\mathrm{m}, 0.1\,\mathrm{m}]$, and $\sigma_s \in [4, 8]$.

4.3.2. The spatiotemporal joint-bilateral geometry filter

Spatial filtering alone cannot suppress noise completely – residual low-frequency noise is still visible in Figure 4.8. However, since this noise is independent for every time step, it can be further reduced by averaging frames from several time steps. The spatiotemporally filtered distance at **p** and time step t is a linear combination of the spatially and temporally filtered distances,

Spatio-temporal filter definition

$$f_{ST}(\mathbf{p}, t) = \varphi \cdot f_s(\mathbf{p}, t) + (1 - \varphi) \cdot f_T(\mathbf{p}, t), \tag{4.5}$$

where the falloff φ specifies the trade-off between spatial filtering and temporal filtering, with $f_T(\mathbf{p}, t)$ propagating filtered distances from the previous time step $t{-}1$ to the current time step t using motion compensation. The larger φ, the more weight is given to the current frame. The results reported here use $\varphi \in [0.01, 0.1]$.

A basic technique such as exponential averaging of the spatially filtered distances over a window of time generates artefacts in areas of high motion. This filters pixels which are not in correspondence, for example across depth discontinuities, and leads to 'smearing' artefacts. To address this problem, previous frames need to be motion-compensated to align moving objects across frames.

Temporal smoothing

previous frame ($t{-}1$)   current frame (t)
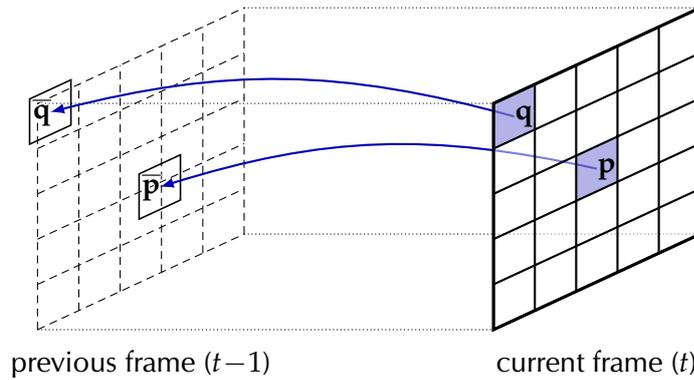
Figure 4.6: Illustration of the motion-compensated filter kernel: arrows indicate optical flow from the kernel centre **p** and a kernel pixel **q** in the current frame (t, right) to the previous frame ($t{-}1$, left).

Let $\overline{\mathbf{p}}$ at time step $t{-}1$ denote the motion-compensated location of **p** at time step t, as shown in Figure 4.6. The temporal contribution towards **p** is computed by

Definition of temporal smoothing

$$f_T(\mathbf{p}, t) = \sum_{\mathbf{q} \in N_{\mathbf{p}}} w(\mathbf{p}, \mathbf{q}, \overline{\mathbf{p}}, \overline{\mathbf{q}}) \cdot f_{ST}(\overline{\mathbf{q}}, t - 1), \tag{4.6}$$

which combines distances from the previous spatiotemporally filtered distance map at the motion-compensated locations $\overline{\mathbf{q}}$ of all pixels **q** in the filter kernel $N_{\mathbf{p}}$. This drastically reduces the flickering caused by time-independent noise.

The filter weights $w(\mathbf{p}, \mathbf{q}, \overline{\mathbf{p}}, \overline{\mathbf{q}})$ in Equation 4.6 are designed to evaluate the similarity of the motion-compensated pixel $\overline{\mathbf{q}}$ to the centre pixel **p**:

Filter weights

$$w(\mathbf{p}, \mathbf{q}, \overline{\mathbf{p}}, \overline{\mathbf{q}}) = w_c(\mathbf{p}, \overline{\mathbf{q}}) \cdot w_d(\mathbf{p}, \overline{\mathbf{q}}) \cdot w_s(\overline{\mathbf{p}}, \overline{\mathbf{q}}) \cdot w_f(\mathbf{q}, \overline{\mathbf{q}}). \tag{4.7}$$

As before, the first three weights determine similarity in colour and distance, as well as spatial proximity. However, the spatial weight w_s does not penalise distance from **p**, but its motion-compensated location $\overline{\mathbf{p}}$.

Flow weight Finally, the flow weight w_f reduces the influence of past data in areas of fast motion, as they tend to be unreliable:

$$w_f(\mathbf{q}, \overline{\mathbf{q}}) = \exp\left(-\|\mathbf{q} - \overline{\mathbf{q}}\|^2 / 2\sigma_f^2\right). \tag{4.8}$$

For the prototype camera described in Section 4.1, $\sigma_f \in [4, 5]$ produces good results.

Handling fast motions Fast motion also leads to increased noise levels in time-of-flight distance maps, as multiple images are sampled per frame (Kolb et al., 2010). To suppress this noise in areas of fast motion, the spatial filter is augmented by redefining $w_c(\mathbf{p}, \mathbf{q})$:

$$w_c(\mathbf{p}, \mathbf{q}) = \exp(g_c) \cdot G_{\sigma_c}(\|\mathbf{i}(\mathbf{p}, t) - \mathbf{i}(\mathbf{q}, t)\|), \tag{4.9}$$

$$g_c = \left[2 - \|\overline{\mathbf{q}} - \mathbf{q}\| / \sigma_f\right]_0^1, \tag{4.10}$$

where $[x]_a^b$ clamps x to the range $[a, b]$. The result is that in areas of fast scene motion, the importance of spatial filtering is increased and distances are smoothed across colour edges. In practice, this effectively prevents motion noise amplification.

Implementation I use a full-kernel implementation of the spatiotemporal filter, as the non-Gaussian colour weight w_c cannot be easily mapped to the acceleration approaches discussed in Section 2.5.3. The correspondences across frames are computed using Brox et al.'s optical flow (2004), adapted and implemented for GPUs by Eisemann et al. (2008).

4.4. Results

The prototype camera was used to record a variety of scenes, including close-ups of objects, close-ups of people, and multiple people interacting. A total of 26 sequences were captured, with over 30 000 frames and about 35 minutes in length. The RGBZ video processing pipeline described in this chapter produces good results with remaining artefacts only in very few frames. The captured sequences are used to demonstrate the efficacy of the proposed processing techniques.

Recorded sequences

The proposed geometry fill-in step removes and fills in unreliable and half-occluded geometry, as demostrated in Figure 4.4. While similar results can be achieved by a single joint-bilateral filter with a larger kernel size, the proposed multi-resolution approach is $5-6\times$ faster, while outperforming a smaller kernel with similar run time in terms of quality.

Geometry fill in

The proposed spatiotemporal filtering step computes visually plausible high-quality distance maps from spatially and temporally noisy input data. In contrast, a simple spatial filter still exhibits noise and flickering, which are effectively removed by the spatiotemporal method (see comparison in Figure 4.7). Figure 4.8 shows that the spatiotemporal filter clearly produces the best results: the distance maps are free of noise, have a higher spatial resolution and clean object boundaries. However, the full improvement only becomes apparent in motion, in the supplementary video[19].

Geometry filtering

The overall filtering approach is also applicable to the Microsoft Kinect without algorithmic modifications – only two filtering parameters were tweaked. Although the Kinect has different noise characteristics than time-of-flight cameras, similar problems exist, like the disparity between depth maps and video. The last row of Figure 4.8 shows that the filtering approach is similarly necessary and leads to clearly improved, coherent depth maps, with the Kinect's typical depth quantisation steps smoothed out.

Microsoft Kinect

spatially-filtered distance maps   spatiotemporally-filtered distance maps
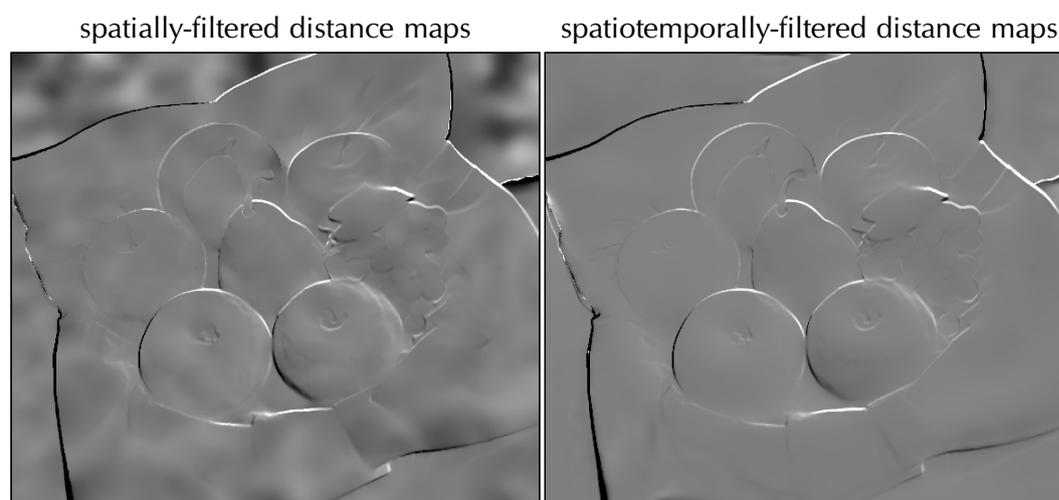

Figure 4.7: Difference images of consecutive distance maps using spatial and spatiotemporal filtering (scaled to show differences of ± 2.5 cm). Motion creates black and white boundaries around objects, but temporal noise results in cloudy deviations from grey (see left image).

---

[19] http://richardt.name/rgbz-camera/

| no filtering (just aligned) | spatial filtering only | spatiotemporal filtering |
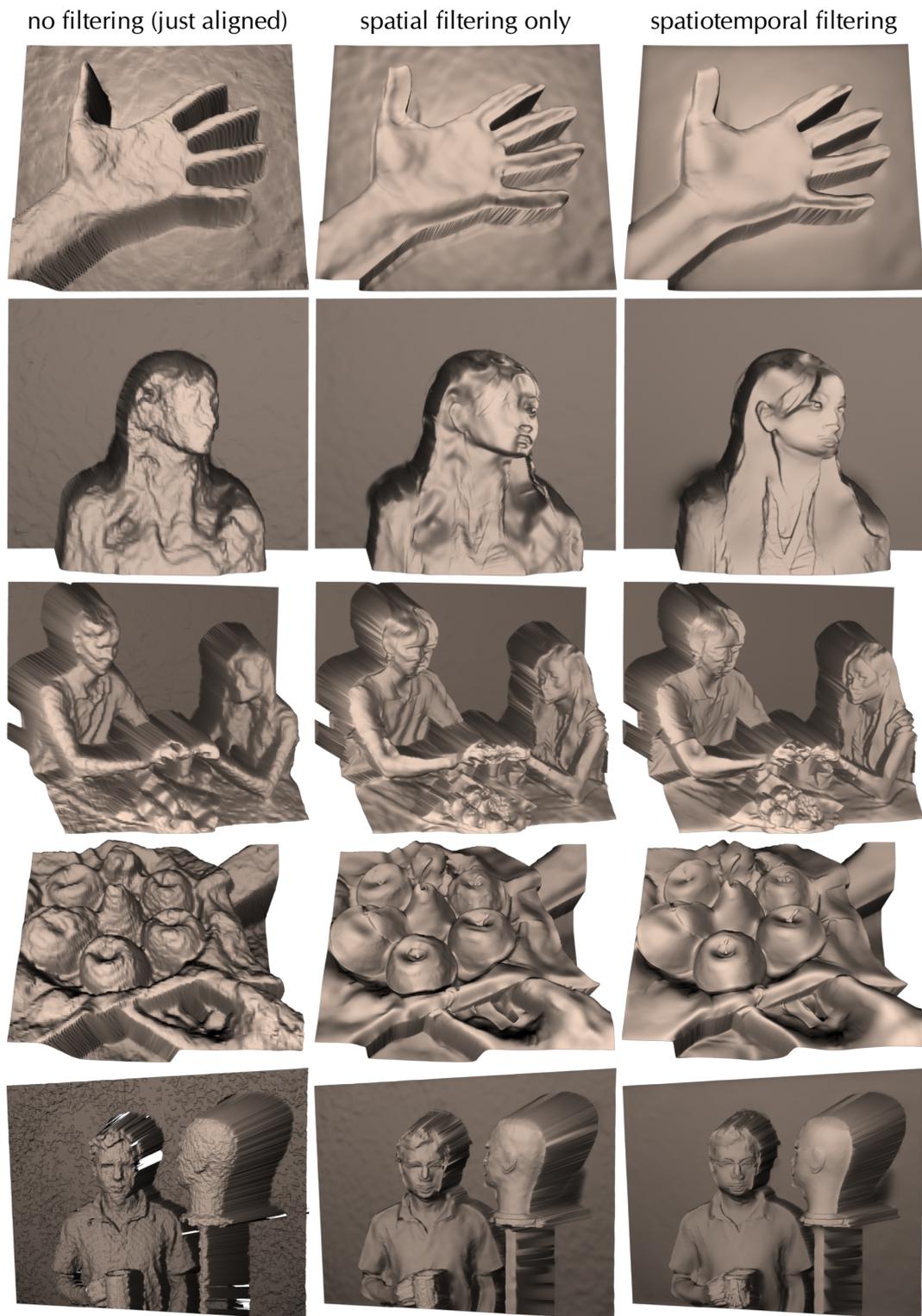
Figure 4.8: Mesh renderings of distance maps without filtering, with spatial and spatiotemporal filtering. It is essential to refer to the supplementary video to see the full improvement in quality. The last row shows data captured using a Microsoft Kinect, with holes and the characteristic depth quantisation steps. This chapter's filtering approach results in a clear improvement.

4.4.1. Run time and performance

Run times

The RGBZ video processing runs at a frame rate of 5.2 Hz for a 584×506 RGBZ video in the prototype implementation – which uses a GeForce 295 graphics card and a 2.8 GHz quad core processor. The bottleneck is GPU time, which divides as follows: 28 per cent each for computing optical flow and the spatiotemporal filter, and 22 per cent each for the geometry fill-in and view alignment computations. The RGBZ video processing can be used interactively, which allows processing and also application parameters to be modified on-the-fly and their outcome to be observed. Alternatively, the video processing can be performed in an offline pre-processing step, and the filtered video can then be used in real time. Additional fine-tuning and advances in hardware will soon make end-to-end real-time processing feasible.

Comparison to previous work

This work is inspired by the work of Snavely et al. (2006), but it differs in some important aspects. The entire RGBZ processing pipeline is specifically designed for interactive performance, whereas their reconstruction and registration steps are computationally expensive and hence not suited for interactive processing of input data. The proposed multi-resolution fill-in procedure also produces higher quality distance maps than the simple interpolation used by Snavely et al. An important consideration in the design of the filtering pipeline was to handle the challenging noise characteristics present in recent depth cameras. Due to this, the proposed approach produces higher-quality, noise-free distance maps at interactive frame rates, whereas Snavely et al.'s results still exhibit flickering due to temporal noise.

4.4.2. Limitations

Optical flow

The result of the spatiotemporal filter depends on the quality of the optical flow between frames: in areas of fast motion, optical flow tends to be unreliable due to motion blur, large displacements and occlusions, which can lead to artefacts such as smearing in the filtered distance map. This could be ameliorated by using a higher capture frame rate, as this reduces the maximum extent of motions, or by extending the optical flow to also respect depth discontinuities.

Colour/depth edges

The geometry fill-in and filtering steps rely on the assumption that similar colours imply similar depth. This assumption can be violated in two ways: (1) strong texture on smooth geometry may introduce 'texture copy' artefacts into the distance map; and (2) depth edges with small colour differences may not be preserved well, for example at the left shoulder in the second row of Figure 4.8. Nevertheless, this assumption holds in many real scenes.

Depth detail

The used time-of-flight camera also has a low spatial resolution, which limits the level of detail in the filtered distance map. Although the filtering increases the resolution beyond the physical limits of the depth sensor, not all fine details can be recovered. Even more detail may potentially be recovered by refining the result using shape-from-shading, but this would incur additional computational costs.

Joint-bilateral filter

The joint-bilateral filtering approach taken in Section 4.2 is not guaranteed to fill geometry optimally, as new values are computed as a linear combination of existing values. This may result in incorrect geometry in the geometry fill-in and spatiotemporal filtering steps. In practice, this only has limited influence on the quality of reconstruction, but it allows the system to run at interactive frame rates.

4.5. Conclusion

RGBZ video camera This chapter presented the first computational camera system that captures high-resolution coherent RGBZ videos at interactive frame rates. It builds on a prototype camera which combines a colour video camera with a recent time-of-flight camera, and an RGBZ video processing pipeline that turns the noisy, low-resolution distance maps and the high-resolution colour video into a plausible and coherent high-resolution RGBZ video.

Pipeline stages The proposed RGBZ video processing pipeline consists of three processing stages:

1. the depth video is aligned to the colour video using reprojection (Section 4.1),

2. unreliable geometry is invalidated and filled in again (Section 4.2), and

3. a spatiotemporal filter gently smoothes the geometry (Section 4.3).

The previous section showed qualitative evidence that these steps work effectively and efficiently.

Discussion Time-of-flight cameras are limited in their spatial resolution and the accuracy of their distance measurements. New models will no doubt feature increased sensor resolutions, and perhaps even improved depth accuracy, but they will be behind commercial video cameras in terms of resolution for years to come – if they ever catch up, which I doubt. I believe that time-of-flight cameras will remain noisy by nature, and the mismatch in sensor resolution between time-of-flight and video cameras will remain a problem. The proposed geometry filtering approach hence remains valid.

Next chapter The next chapter demonstrates that the RGBZ videos created using the filtering approach described in this chapter enable a wide variety of video processing effects which are unobtainable from videos alone.

RGBZ VIDEO PROCESSING EFFECTS

5

Over millions of years, the human visual system has evolved to process visual inputs to make sense of the world around us. The geometry of objects and their arrangement in space is integral to our experience of the world. Hence, the human visual system has evolved to elucidate the spatial relationships between objects in a multitude of ways (for depth perception see Section 2.2). It is this inference of geometric information that enables our visual system to perform so well.

Introduction

Similarly, many video processing effects are infeasible from video input alone, and require additional geometric information. The lighting of objects, for example, is principally determined by surface normals. To synthesise new views from different viewpoints, one also needs to know the depth of objects to render them in the correct order and with correct parallax. In addition to this, depth is also an important semantic cue, as nearby objects are generally more salient than distant ones.

Motivation

However, existing video cameras cannot capture geometric data. In the preceding chapters, I therefore considered two approaches to additionally acquire geometric information. The first approach, in Chapter 3, adds a second video camera to infer scene depth using stereo matching – one purely passive geometry capture approach surveyed in Section 2.3. However, the quality of the computed disparity maps is insufficient to achieve some of the effects presented in *this* chapter, because of poor depth accuracy and hence inaccurate surface normals. The approach in Chapter 4 instead uses an additional depth sensor in combination with a specialised filtering and upsampling approach that results in high-quality RGBZ videos.

Acquisition of RGBZ videos

Structure of this chapter In this chapter, I present five video processing effects that critically rely on the high quality of geometric information available through RGBZ videos to achieve results that are unobtainable from a colour video alone. Perhaps the most basic effect is background segmentation (Section 5.1), which is greatly simplified when geometry is available. Next, I describe a relighting technique which relies solely on plausible surface normals (Section 5.2). The two following effects extend video abstraction and stroke-based rendering techniques to make use of the geometry for placing lines and brush strokes along meaningful geometric features (Section 5.3). And the last effect renders videos in stereoscopic 3D by synthesising new views (Section 5.4).

Performance The effects shown in this chapter all work at real-time frame rates (15 Hz or more). An interesting and powerful option is to interactively apply these effects subsequent to the RGBZ video processing pipeline in Chapter 4. This allows processing and rendering parameters to be modified on the fly and their outcome to be observed – all at interactive frame rates (about 4 Hz).

5.1. Video foreground segmentation

Perhaps the most basic use of depth information is 'z-keying': thresholding depth to separate the foreground from the background. I implemented a slightly more general technique that thresholds each pixel relative to a 3D plane in world space. Given a pixel's coordinates $\mathbf{p} = (x, y, z, 1)$, and a plane defined in the form $\mathbf{n} \cdot \mathbf{p} = (a, b, c, d) \cdot (x, y, z, 1) = ax + by + cz + d = 0$, their distance is $\delta = \mathbf{n} \cdot \mathbf{p} / \|(a, b, c)\|$, where the plane's normal vector (a, b, c) lies in the half-space with positive distances.

Segmentation

This is particularly useful when the background in a video is not orthogonal to the camera's view direction, as is often the case. For example, the fruit bowl sequence in Figure 5.1 was recorded at an angle above a table and z-keying is hence unable to segment it. However, using an oblique plane 7 cm above the table's surface works well, as demonstrated in the figure.

Oblique planes

Thresholding is a binary operation which classifies pixels to be either visible or not, leading to a binary matte with hard boundaries. These boundaries can easily be softened using a 3×3 Gaussian blur. An additional matting step would likely further improve results by separating foreground and background colour cleanly, but this would come at additional computational expense.

Basic matting

Figure 5.1 shows an example which compares the segmentation results based on the unfiltered and spatiotemporally filtered distance maps, respectively. It is clear that the foreground is not cleanly segmented from the background when using the unfiltered distance map. On the other hand, using the filtered distance map produces a cleanly segmented foreground.

Example
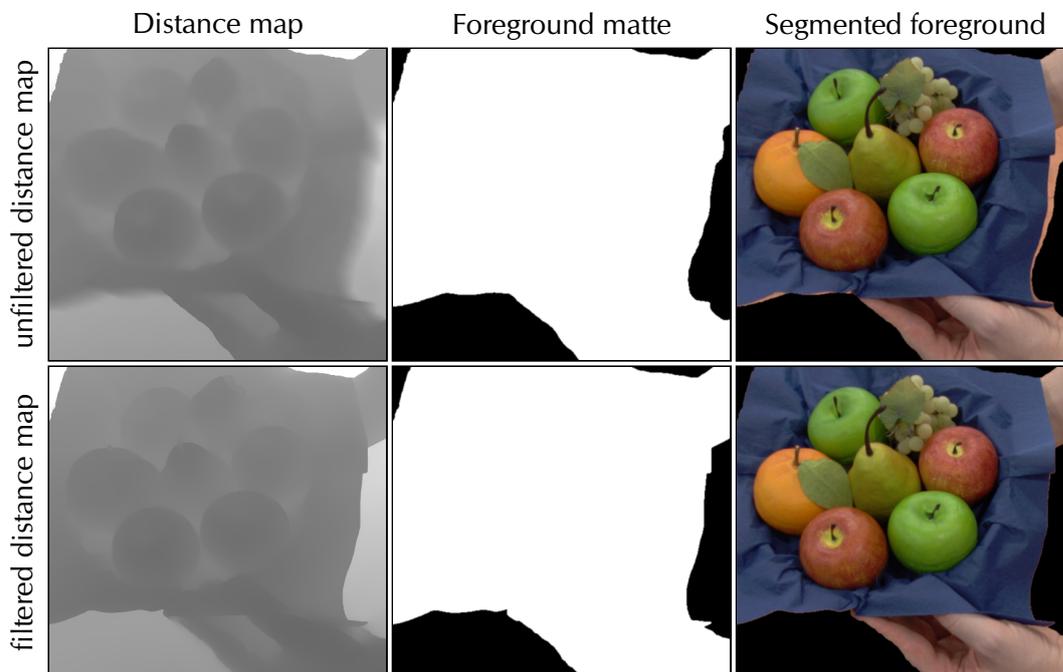
Figure 5.1: Examples of video foreground segmentation. The filtered depth produces clean object outlines.
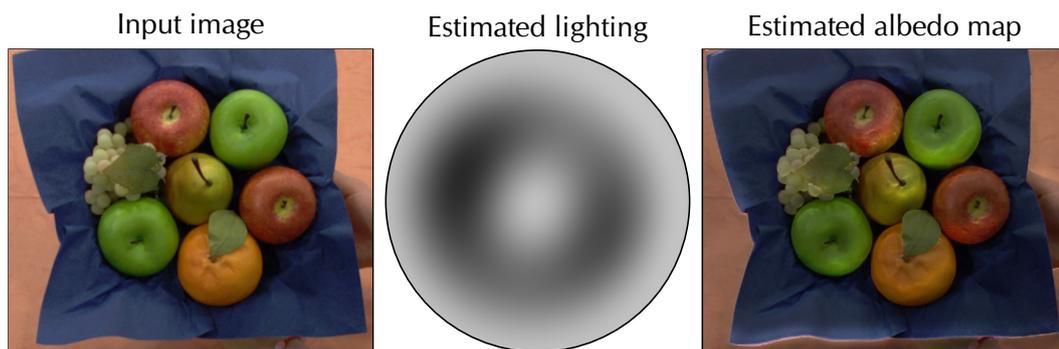
5.2. Video relighting

Motivation Photographers and cinematographers routinely use lighting to great effect in order to achieve a desired aesthetic effect, such as setting the mood or directing attention. In these settings, lighting is often carefully designed and professionally set up, and it cannot be changed easily in post-production unless a suitable lighting basis has been captured in a light stage (Wenger et al., 2005). However, this is impractical for casual video camera users. Alternatively, a still image can be relit if a specific lighting model is assumed and scene geometry is known, for example if a normal map is painted by the user (Okabe et al., 2006). This is clearly infeasible for RGB videos, but RGBZ videos provide the required geometry.

Approach Using existing methods, we[20] implemented a simple technique for relighting RGBZ videos in real time, which consists of three steps:

1. estimation of scene illumination from a single RGBZ video frame,

2. computation of albedo maps for each video frame, and

3. rendering of the albedo map with new lighting parameters.

Figure 5.2 shows the main components of our video relighting approach.

| Input image | Estimated lighting | Estimated albedo map |

Figure 5.2: The main components of our video relighting: the input image is used to estimate incident illumination, which is then removed from the input image to compute an albedo map.

Lighting estimation The scene illumination is assumed to be constant and thus estimated offline from a single RGBZ video frame, using the known surface normals and corresponding pixel colours. By assuming purely diffuse, Lambertian reflectance, the estimation of the incident lighting reduces to computing the first 9 spherical harmonic coefficients of the environment map (Ramamoorthi and Hanrahan, 2001a; Basri et al., 2007). Only the most reliable pixels are used for the lighting estimation: pixels at or near depth discontinuities or in shadows are manually masked out. Without a given albedo map, lighting estimation is ambiguous, as the colour of an object depends on both its albedo and the colour of the lighting. Hence, we estimate the lighting independently for the red, green and blue colour channels, and the one with the strongest constant coefficient is taken to be the scene illumination. The albedo map is then computed by dividing each pixel's colour by the estimated illumination.

---

[20] Chenglei Wu (MPI Informatik) created the original implementation of this technique and I translated the relighting component to GLSL shaders for real-time video relighting.

In principle, any lighting model could be applied to the computed albedo maps to relight the original video. We use the same spherical harmonics technique for rendering new lighting conditions, as it efficiently approximates environment maps for diffuse objects (Ramamoorthi and Hanrahan, 2001b). Figure 5.3 shows several plausible relighting results created using this approximation to light probes (inset). *Relighting*

Figure 5.3: Example frames from relit video sequences. The light probes are courtesy of Paul Debevec.

5.3. Non-photorealistic rendering of videos

Introduction Previous non-photorealistic rendering techniques for videos produce effects based on imagery alone (Collomosse et al., 2005; Winnemöller et al., 2006). The aim of the work described in this section is to show that the depth data in RGBZ videos enables more advanced non-photorealistic video rendering effects.

5.3.1. Related work

NPR camera Raskar et al.'s non-photorealistic camera (2004) was the first work to reveal the potential of depth in non-photorealistic rendering. They built a prototype camera that uses four flashes to detect and highlight depth edges in real-world scenes. This effectively highlights occlusion boundaries which helps in conveying shape features. However, no actual scene geometry is recovered apart from the location and orientation of the depth discontinuities.

2.5D video The first non-photorealistic rendering technique for RGBZ videos is due to Snavely et al. (2006), who describe how to process and stylise '2.5D videos'. The source of these videos is an existing active spacetime stereo approach (Zhang et al., 2003). The raw depth maps are first filtered bilaterally and shape correspondence between frames is then optimised, to achieve coherent stylisation. As temporal information is not taken into account during the filtering step, some temporal noise remains.

RGBN images The first real-time photometric stereo system for capturing images with normals ('RGBN images') was demonstrated by Malzbender et al. (2006). Their system uses a high-speed video camera that captures video frames illuminated by up to 16 LEDs. They compute surface normals on the GPU using photometric stereo (Section 2.3), and then perform basic reflectance and normal transformations to emphasise the surface detail of objects, for example for documenting archaeological artefacts.

RGBN stylisation Toler-Franklin et al. (2007) go into more detail on how to process and stylise RGBN images. They describe tools for filtering, curvature estimation and segmentation of RGBN images. However, the largest part of their paper concentrates on showing a large variety of rendering styles that can be applied to such RGBN images: from straightforward cel shading and suggestive contours (DeCarlo et al., 2003) to the more complex multiscale curvature shading style.

Context-aware light source Wang et al. (2010) also use photometric stereo to compute real-time normals, with the aim to transfer a new style onto the scene by projecting the appropriate image. To achieve their goal, they use a beam splitter to align an infrared camera – for photometric stereo – with a projector that projects scene-enhancing imagery, for example to sharpen image features in a scene. However, as no motion compensation is applied, any scene motion results in incorrect normals.

Bottom line Apart from the last, all of these systems interfere with the scene in the visible light spectrum. In contrast to this, the prototype camera described in Chapter 4 captures colour and depth simultaneously without interference. A further disadvantage of the majority of described systems is that photometric stereo does not provide depth information per se, but just normal maps. These would need to be integrated to compute depth maps, which amplifies any errors in the normal map.

5.3.2. Geometry-based video abstraction

Natural scenes are cluttered with unnecessary detail and, as a response, the human *Motivation*
visual system has evolved to find the most visually salient regions on which to
focus our attention. Yet, abstracting imagery by smoothing low-contrast regions
and emphasising high-contrast features can further boost the recognition rate of
faces and performance in memory tasks (Winnemöller et al., 2006). Similarly, line
drawings, which take abstraction to the extreme, can effectively depict shape, with
computer-generated line drawings rivalling the effectiveness of artists' drawings
(Cole et al., 2009).

Using RGBZ videos, it is possible to unify video abstraction and line drawings (Sec- *Approach*
tion 2.1.1) into a geometry-based video abstraction rendering style that combines
the benefits of both original styles. This new style consists of three components,
which are illustrated in Figure 5.5: an abstracted colour video, toon shadows and
abstraction lines. These are composited to produce the final result.

The colour video is abstracted using a bilateral filter (Section 2.5), with parameters *Abstracted colour*
$\sigma_s = 3$ and $\sigma_r = 0.1$, which gently smoothes low-contrast regions while preserving
edges in the image. Afterwards, I increase saturation by 40 per cent using Haeberli's
method (1990), to make the colours more vibrant:

$$\text{luminance} = (0.2126, 0.7152, 0.0722) \cdot (r, g, b), \tag{5.1}$$

$$\text{greyscale} = (\text{luminance}, \text{luminance}, \text{luminance}), \text{ and} \tag{5.2}$$

$$\text{result} = [(1 - saturation) \cdot \text{greyscale} + saturation \cdot (r, g, b)]_0^1, \tag{5.3}$$

which uses the operator $[x]_a^b$ to clamp the value of x to the range $[a, b]$.

The second component is a cel-shaded version of the scene geometry which exhibits *Toon shadows*
toon shadows with prominent, stylised shadow boundaries. I use the following
sigmoid *toon step* function (see Figure 5.4) to map diffuse shading to toon shading:

$$\text{toon shading} = \tau_o + \tau_s \cdot S(\tau_l, \tau_h, \text{diffuse shading}), \tag{5.4}$$

where S is the `smoothstep` function[21], and the default parameters are $\tau_o = 0.7$ (for
offset), $\tau_s = 0.3$ (for scale factor), $\tau_l = 0.4$ and $\tau_h = 0.6$ (for low and high end of step).
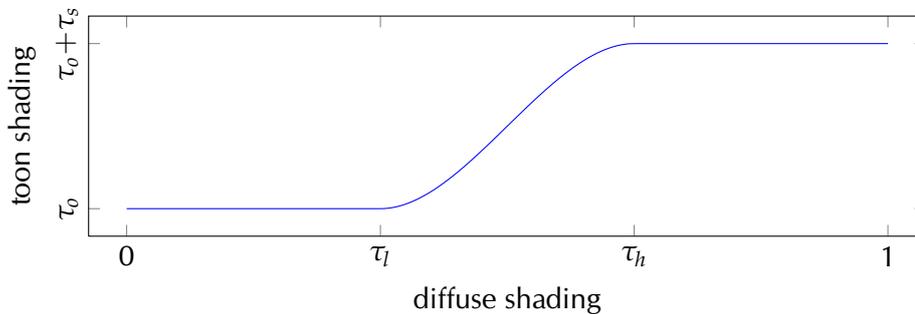

Figure 5.4: Plot of the toon step function in Equation 5.4.

---

[21] A common implementation of the `smoothstep` function is $S(a, b, x) = 3t^2 - 2t^3$ with $t = \left[\frac{x-a}{b-a}\right]_0^1$.
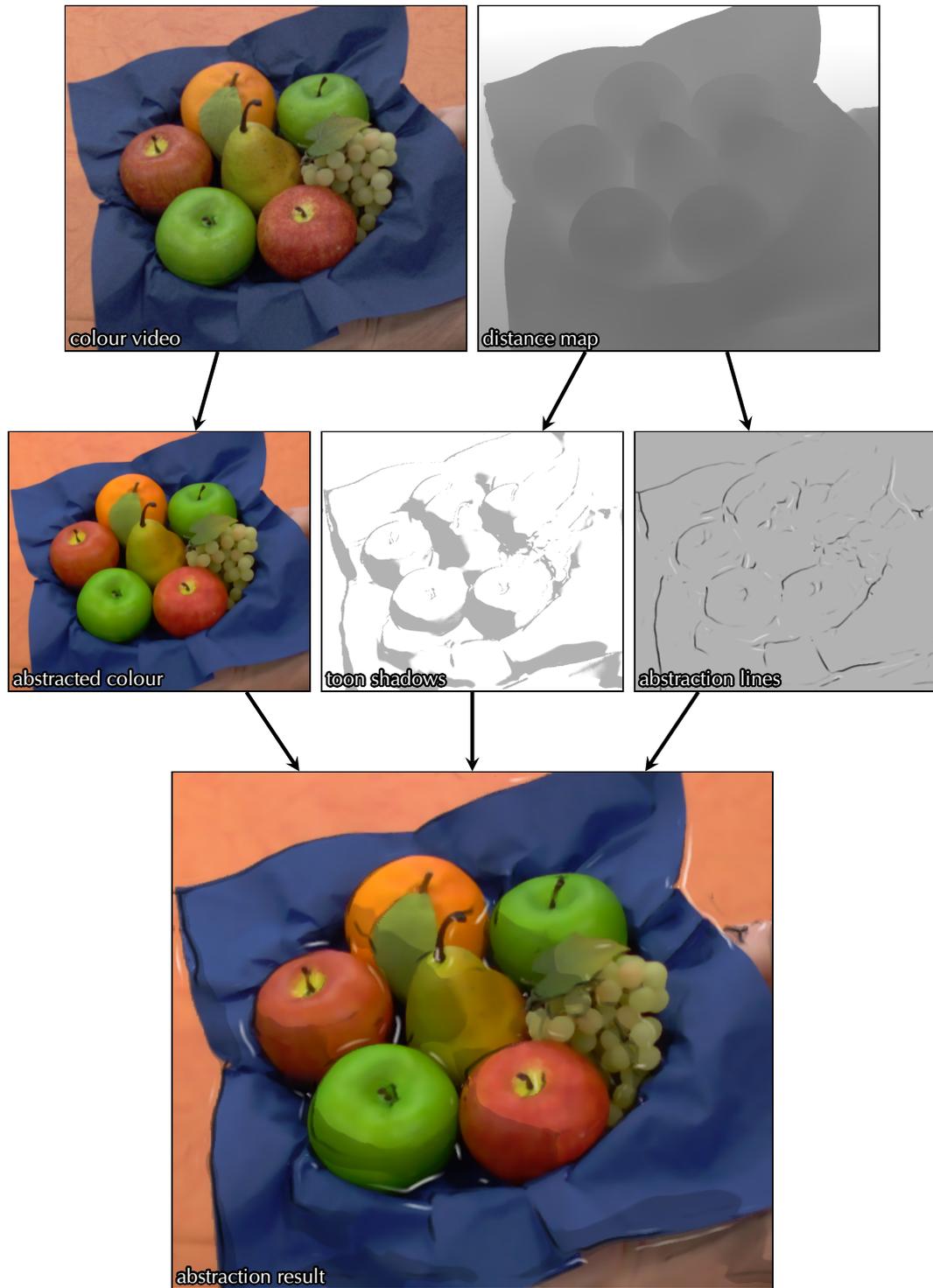
Figure 5.5: The main components of my geometry-based video abstraction style.

The final component uses Lee et al.'s image-based line drawing technique (2007), *Abstraction lines* which detects ridges and valleys in a diffusely shaded image of the scene geometry. These lines help to visually communicate the shape of objects in the scene above and beyond what can be derived from the colour video. Please refer to Lee et al.'s paper for a description of the algorithm and its parameters. The default settings I used are: a line width of $w = 8$, a step size of $\beta = 0.3$, and lower and upper curvature thresholds, respectively, of $c_l = 0.01$ and $c_u = 0.05$; however, most parameters are tweaked to produce appealing results on each individual video sequence.

The three components of the geometry-based video abstraction style are composed *Composition* by multiplying the abstracted video (with an exponential weight of 0.6) and toon *& results* shadows, and adding the black and white lines using alpha blending – the line opacity determining the amount of black or white used. Figure 5.6 compares the result to Winnemöller et al.'s purely image-based approach, which relies only on image information to create an abstracted video. Additional results for more RGBZ videos are shown in Figure 5.7.

Winnemöller et al.'s abstraction   geometry-based video abstraction
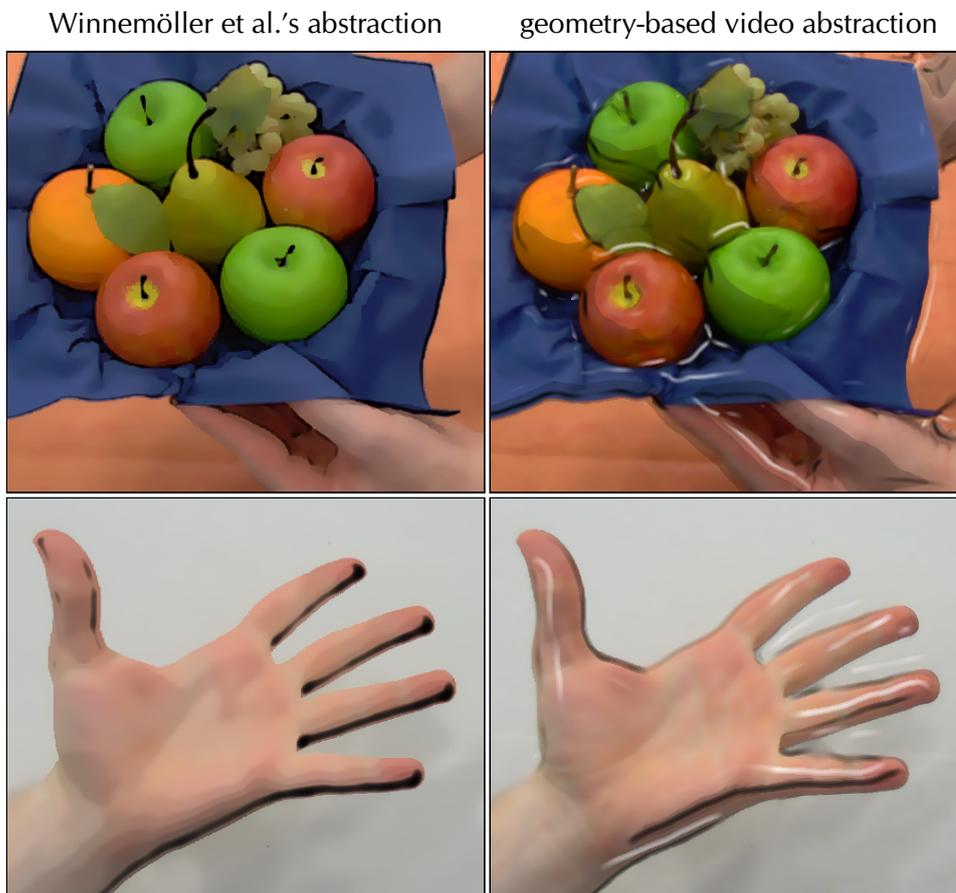
Figure 5.6: Comparison to Winnemöller et al.'s video abstraction technique. Since the proposed technique exploits scene geometry, it places feature lines at geometrically meaningful locations.

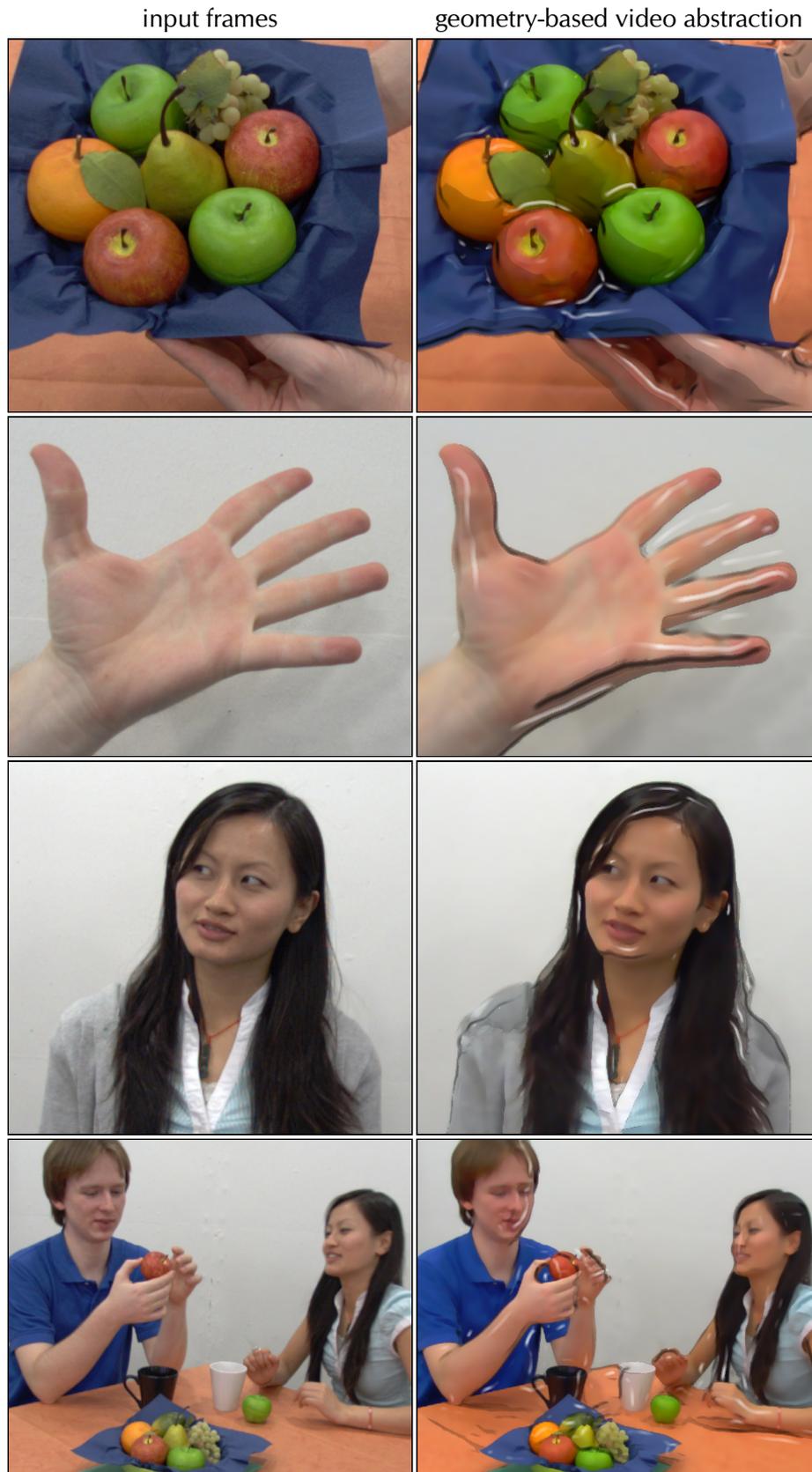input frames   geometry-based video abstraction

Figure 5.7: Examples of geometry-based video abstraction.

5.3.3. Stroke-based rendering

Many artistic techniques in the visual arts essentially come down to the skilful placement of primitives such as paint strokes, hatches or stipples on a medium such as canvas, wood or paper. Artists rely on a wealth of experience as well as geometric scene knowledge to help them with stroke placement. In recent years, many NPR techniques have been proposed that imitate these stroke-based rendering styles (Hertzmann, 2003), but they all rely on either only image content or only geometric models. Here, I propose a technique that uses both colour and depth data. *Motivation*
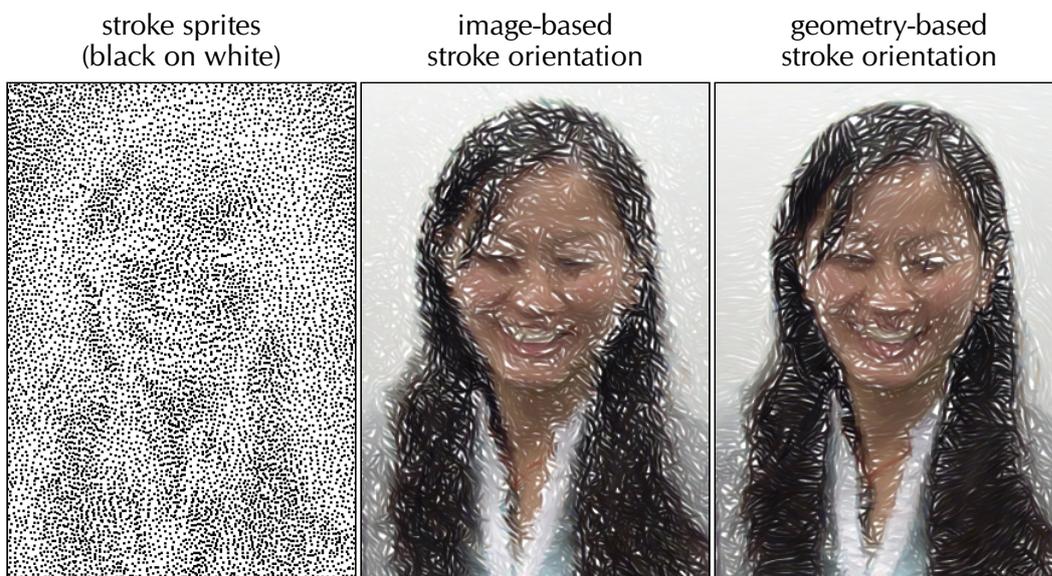
The technique is based on the stochastic sprite distribution approach by Lu et al. (2010), which generates sprites (one for each stroke) that uniformly cover the image area. The density of sprites is calculated using an off-screen coverage buffer into which each sprite is rendered additively as a square, so that the value of each pixel will be the number of sprites that overlap with it. To maintain a uniform density of sprites, new sprites are added and old sprites deleted stochastically if the coverage falls below or exceeds given thresholds. These sprites are advected from frame to frame using optical flow, to follow the underlying image content. In my implementation, I reuse the optical flow computed as part of Section 4.3.2. *Sprite distribution*

Lu et al. did not name the parameters they use, so I call the size of the square drawn into the coverage buffer the *coverage size* (default size: 4, for a 4×4 square). The *target coverage* determines the desired number of overlapping sprites at each pixel in the coverage buffer (default: one sprite per pixel). New sprites are added with a certain probability if the coverage is less than the *add threshold* (default: 1, for 1×*target coverage*); conversely, existing sprites are deleted if the coverage exceeds the *delete threshold* (default: 3, for 3×*target coverage*). The probabilities of adding and removing sprites depends on the difference between the current and *target coverage*, so that the coverage after this step will be close to the *target coverage*. *Distribution parameters*

stroke sprites (black on white)	image-based stroke orientation	geometry-based stroke orientation

Figure 5.8: Position of sprites (left) and comparison between image- and geometry-based stroke orientation. The geometry-aligned strokes represent the face better than strokes aligned to image gradients.

Stroke orientation Lu et al. orient strokes along image gradients; however, using the distance map, the strokes can be aligned along meaningful geometric features instead (Figure 5.8). In this case, strokes are aligned along the principal curvature at the sprite position. Like the original approach, I draw strokes in the screen plane, and not in 3D space, as this produces a more painterly look (Meier, 1996). For additional artistic effect, the position and orientation of strokes can also be jittered.

Stroke rendering The brush strokes can be rendered with different stroke textures, but single-coloured strokes nonetheless often produce appealing results. I use a single layer of elongated strokes of a single *stroke size*, by default 10×2 pixels. The resulting stroke-based rendering style is the most flexible of all effects described in this chapter, as changing the many rendering parameters results in a large variety of stroke-based rendering styles (Figure 5.9).

Stroke attenuation A rendering style that closely resembles hatching can be achieved by attenuating strokes inversely to the diffuse shading at their position, so that strokes concentrate near object boundaries. I attenuate strokes by reducing their opacity using

$$\text{alpha} = \left[1.0 - attenuation \cdot \sqrt{\text{diffuse shading}}\right]_0^1, \tag{5.5}$$

where the *attenuation* factor is in the range $[1, 1.1]$, or zero to disable the effect. Examples of this stroke attenuation style are shown in Figure 5.10.
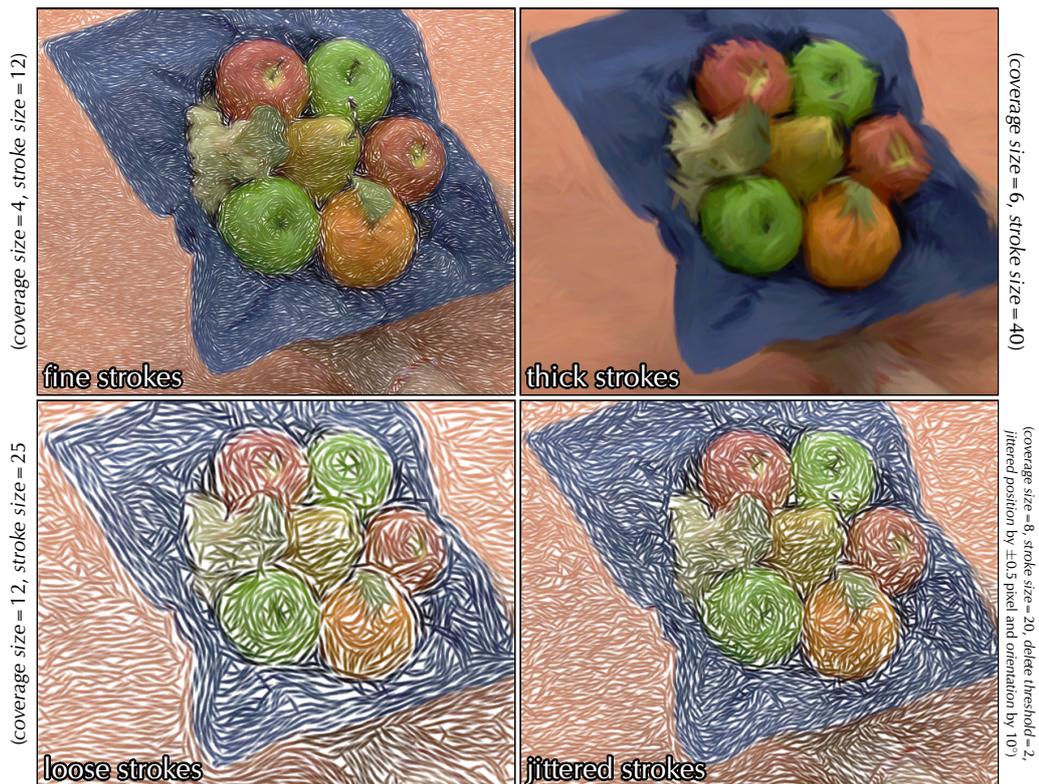

Figure 5.9: Illustration of a variety of rendering styles created by modifying rendering parameters.

input frames   stroke-based rendering   with attenuated strokes

Figure 5.10: Examples of stroke-based rendering with and without stroke attenuation. The attenuated examples also use the foreground segmentation to remove the background (Section 5.1).

5.4. Stereoscopic 3D rendering

Motivation The previous RGBZ video effects have demonstrated a few different ways in which the depth component of RGBZ video can be used to improve on conventional video processing effects. However, all these effects only capture a single view of the scene. This section shows how RGBZ videos can be rendered stereoscopically to improve the perception of scene depth (Section 2.2). In a sense, rendering the RGBZ videos stereoscopically makes their depth maps perceivable by humans.

Virtual stereo camera setup The key to stereoscopic rendering is to synthesise two views – one for each eye. RGBZ videos make this easy, as they can be reprojected as textured triangle meshes, like in the video alignment step of the previous chapter (Section 4.1). I use a parallel setup of virtual stereo cameras that are horizontally displaced to either side of the real camera position by half the interocular distance. Next, I shift the cameras' image planes horizontally so that the screen plane (of zero disparity) is at a given depth. The interocular and screen distance are parameters that can be adjusted to provide a comfortable viewing experience.

Disocclusions Any change in viewpoint leads to disocclusions – areas that are occluded in the original view which are visible (disoccluded) from the new viewpoint. In fact, these are the same half-occlusions as described in Section 4.1 and they also align with depth discontinuities in the original view. In contrast to the previous discussion, however, there is no additional information available for the new virtual viewpoints, and thus the fill-in procedure of Section 4.2 cannot be applied. Instead, I detect depth discontinuities in the original view and fill the collocated disoccluded areas with the background colour. This colour is sampled from the side of the depth discontinuity furthest from the camera. The virtual cameras are shifted to either side of the real camera, as this minimises the total size of disocclusions in both viewpoints. Figure 5.11 shows that the accurate depth boundaries produced by my filtering approach are required to dramatically reduce artefacts.

using unfiltered depth map　　　　　using filtered depth map
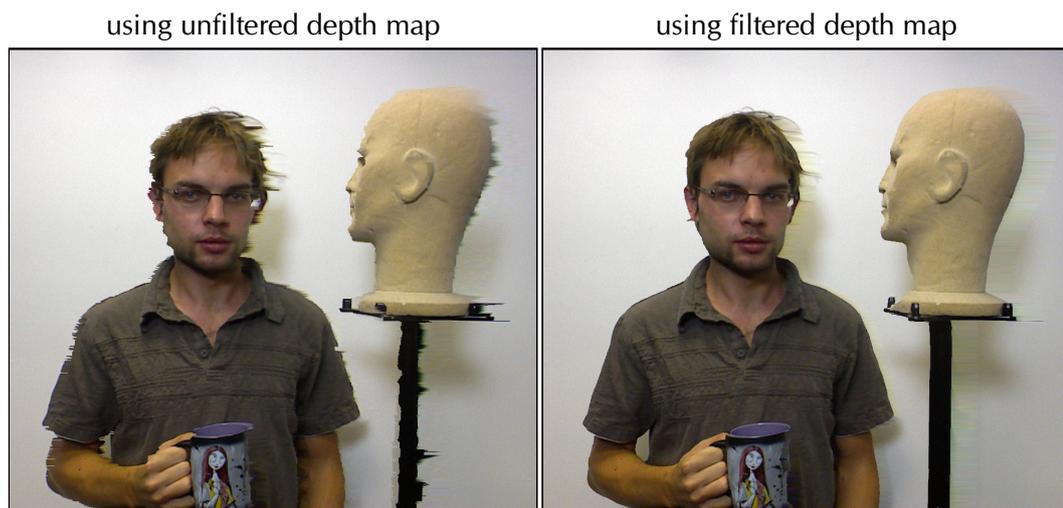
Figure 5.11: The quality of the distance map directly influences the quality of stereoscopic 3D rendering. This is particularly visible in disoccluded regions, as shown in these right stereo half-images. The unfiltered depth data (left) results in many more artefacts than the filtered depth (right).

Using RGBZ videos with reprojection-based stereo view synthesis affords a lot of *Benefits* flexibility as the interocular distance and screen distance can be changed freely *after* recording a video. This allows these setting to be adjusted to produce comfortable yet solid-looking results. Figure 5.13 shows a range of stereoscopic images generated using this approach, shown as half-colour red-cyan anaglyphs.

Unlike traditionally recorded stereoscopic videos, the synthesised videos are RGBZ *Differences* videos themselves, with dense, high-quality depth maps. Depth maps estimated using stereo matching (Section 2.4) are often inaccurate in weakly textured regions whereas the synthesised stereoscopic RGBZ videos do not suffer from this problem. Thus, the RGBZ video processing effects described in this chapter can all be applied in stereoscopic 3D, by simply applying the effects to each stereo view independently.

This is a convenient approach to apply non-photorealistic rendering to stereoscopic *Application:* imagery. Previous work, such as Stavrakis and Gelautz (2004), explicitly warps *stereo NPR* features from one half-image to the other. This is not necessary using the approach proposed here, as shown by the examples in Figure 5.12, which show stereoscopic video relighting and stereoscopic geometry-based video abstraction. However, the stroke-based rendering style would be inconsistent when applied to both stereo half-images, but consistent if warped using Stavrakis and Gelautz's approach.
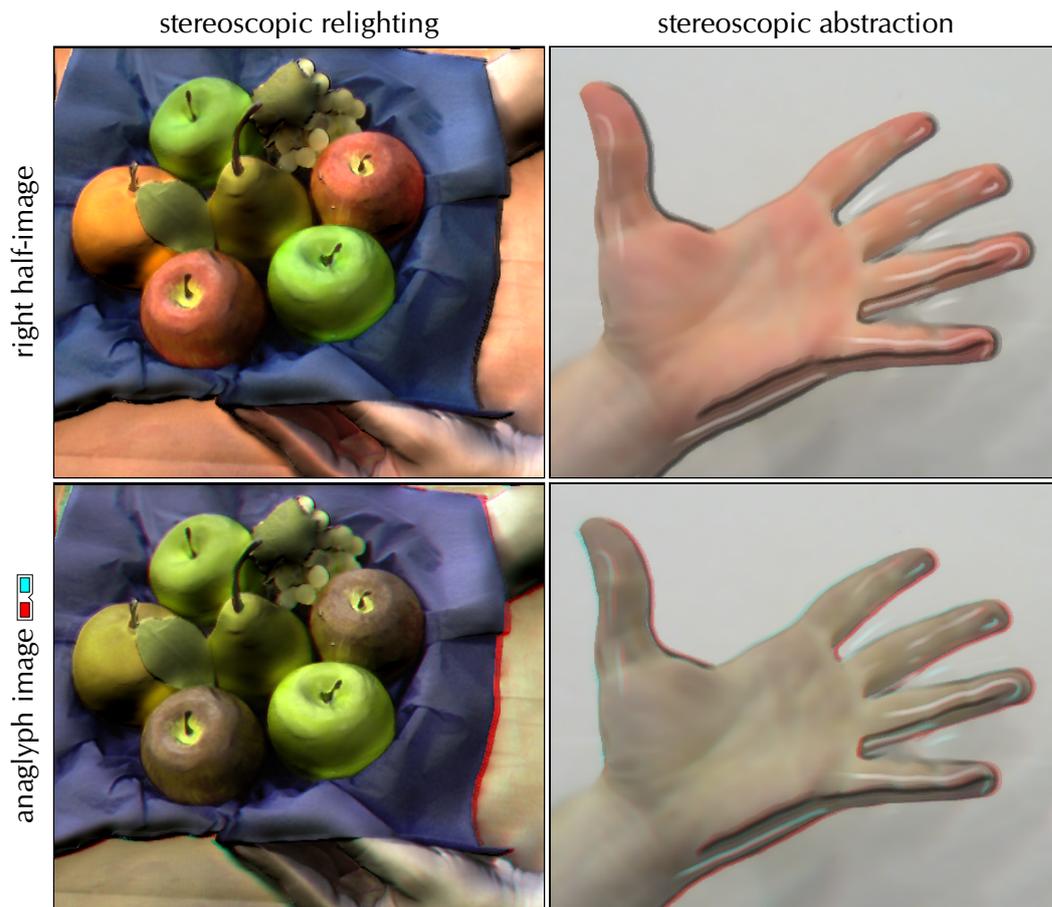
Figure 5.12: Examples of stereoscopic RGBZ video processing effects: video relighting (left) and geometry-based video abstraction (right). The top shows processed versions of the synthesised right half-images and bottom shows the stereoscopic result as half-colour red-cyan anaglyphs ▦▭.

Figure 5.13: Stereoscopic 3D renderings of a range of RGBZ video sequences. The images are shown as half-colour red-cyan anaglyph images ▣▣.

5.5. Conclusion

This chapter has demonstrated that RGBZ videos enable a range of video processing effects that are not achievable from colour videos alone as the geometric information is a critically important ingredient to create these effects. The simplest such example is video foreground segmentation, which produces clean object boundaries from simply thresholding videos relative to a plane.

Usefulness of RGBZ videos

RGBZ videos can also be relit using new lighting environments after they have been captured, and some examples of this were shown in this chapter. However, relighting is most sensitive to geometry inaccuracies and simplifying assumptions, such as diffuse reflectance, and on most scenes, the quality of geometry and lighting estimation was insufficient to produce plausible results.

Video relighting

The availability of geometric information in addition to a colour video also creates new opportunities in video-based non-photorealistic rendering. In this chapter, I described extensions to existing techniques that use the additional depth and normal information. The geometry-based video abstraction style overlays cel-shaded shadows and line drawings extracted from the geometry on top of the conventionally abstracted video for a solid look; and surface normals are used in a stroke-based rendering style to make strokes follow the geometry of objects. These are just two examples; other styles (Section 2.1) could likely be extended as well.

Non-photorealistic rendering

And finally, I presented a technique to create stereoscopic 3D content from RGBZ videos by synthesising the two stereo views. This provides flexibility in stereoscopic rendering, as the view settings can be modified freely to create a comfortable result. The synthesised stereoscopic RGBZ videos can also be used to apply other video effects stereoscopically, by just applying them to both views independently.

Stereoscopic rendering

The video effects proposed in this chapter are temporally coherent by virtue of the temporally coherent RGBZ videos they use as input. However, errors in the filtered geometry, such as blurred depth discontinuities, can still lead to artefacts, such as misplaced lines in the abstraction style, halos in the relit video or boundary errors in the foreground segmentation. Textureless regions in the video further result in poor optical flow, which may lead to correspondence errors becoming visible in some effects, such as strokes that 'swim' over a uniform background. Nevertheless, this chapter shows that RGBZ videos enable computational videography effects of higher quality and complexity than possible before.

Discussion

With algorithmic optimisations and improvements in hardware performance, these video processing effects could soon be applied to video conferencing, for example using the Kinect sensor. By embedding a depth camera and dedicated processing hardware in a digital camera, these effects might also find their way into consumer digital cameras, empowering their users with better and more interesting effects.

Applications

The stereoscopic video effects raise the question of whether applying the same effect to both stereo views always produces a stereoscopic result that is comfortable to look at. The next chapter argues that this is not the case by analysing how various effects cause different human comfort ratings. The aim is a computational model that can predict the level of visual discomfort caused by stereoscopic 3D images.

Next chapter

PREDICTING STEREOSCOPIC VIEWING COMFORT

6

> *This chapter presents research that has been published at CÆ 2011, with the preliminary case study presented at CÆ and NPAR 2010 (Richardt et al., 2010a, 2011). The perceptual study (Section 6.4) was jointly designed and carried out with Ian Davies and Lech Świrski. Lech Świrski further implemented the 'shower door' detection (6.6.2).*

Interest in stereoscopy has seen a resurgence in recent years (Section 2.2.3) – a *Introduction* development primarily driven by the computer gaming and film industries, which are taking advantage of the availability of improved stereoscopic display technology. The primary purpose of all such displays is to present each eye with its own image (or video), so that the human visual system can be tricked into perceiving a scene stereoscopically – see Section 2.2 for an overview of human depth perception.

The transition from standard 2D to stereoscopic computer-generated imagery is *Motivation* not as straightforward as rendering two different viewpoints; care is required to produce visually plausible results that do not cause viewing discomfort. Existing rendering techniques – photorealistic or not – therefore need to be reviewed and revised if necessary to ensure that they work correctly and comfortably in stereo. Non-photorealistic rendering techniques often use randomness to create a hand-crafted look, which causes inconsistencies between views in stereoscopic rendering. A notable exception is the work by Marković, Stavrakis and Gelautz (Section 2.1.3), who explicitly enforce consistency between views.

The primary aim of this chapter is therefore to advance the understanding of how to *Aim* extend non-photorealistic rendering techniques to stereoscopic rendering without causing visual discomfort.

The first step in this direction is a case study, which compares the stereo coherence – *Case study* or consistency of stereoscopic views – in animations created using two stereoscopic variants of the same watercolour rendering style. The result of the case study is clear-cut: the majority of participants preferred the object-based technique over the image-space technique. However, image-based NPR techniques are often more flexible than object-based ones (see Section 2.1), as they require less input: a single image is often sufficient instead of a complete geometric model.

Assessment of visual comfort

More importantly, the case study highlighted the need for objective assessment of stereoscopic viewing comfort. However, as this is very much a subjective – if not subconscious – issue of visual perception, the assessment of visual comfort is usually carried out by expert viewers or a panel of naïve viewers (Neuman, 2008). But conducting such subjective assessment is both time-consuming and costly. For this reason, it would be desirable to automatically assess the visual comfort levels of stereoscopic imagery using a computational model of the human visual system.

Computational model

The main contribution in this chapter is the first computational model for objectively assessing the visual comfort of stereoscopic imagery. The model is based on recent work in visual perception, which showed similarities between human observers and normalised cross-correlation – a local stereo correspondence technique. Combined with tools from stereo computer vision, the coherence scores computed by the model strongly correlate with human comfort ratings.

Experimental validation

The experimental validation of the computational model consists of a perceptual study, in which 20 participants were asked to rate the level of comfort for each of 80 stereo images. The results show that the model performs on a par with human observers, and it could therefore be used to automatically assess the visual comfort levels of stereoscopic imagery, without the need to run costly perceptual studies which would be impractical in real-world uses of stereoscopy like movies or games.

Taxonomy of stereo coherence issues

Throughout both the case study and the larger perceptual study, it became apparent that the differences between left and right stereo half-images cause varying types and levels of visual discomfort. Based on the judgment of expert viewers, we identified three broad categories of stereo coherence issues that cause visual discomfort: binocular rivalry, the shower door effect, and randomness.

Coherence tools

This chapter closes with a discussion of how these stereo coherence issues can be automatically identified and localised using a set of computational tools that build on the computational model.

6.1. A case study in watercolour rendering

In recent years, the issue of temporal coherence in non-photorealistic rendering has received significant attention (DeCarlo et al., 2004; Collomosse et al., 2005; Bousseau et al., 2006). Although the non-photorealistic rendering community agrees that temporal coherence is, in general, a desirable property of NPR techniques, it still lacks a clear definition. The main goal is to suppress flickering, which is often caused by applying rendering techniques on a per-frame basis.

Temporal coherence

By analogy, one could argue that similar attention should be given to the issue of stereo coherence, which is the consistency of the two half-images in stereoscopic imagery. In the real world, the two views created on our retinas are projections of the same 3D world and for this reason typically consistent, except for certain physical phenomena such as reflection and refraction. However, this is not necessarily the case for non-photorealistic rendering techniques if it is not explicitly enforced.

Stereo coherence

Image-based NPR techniques, in particular, are prone to introduce inconsistencies when applied stereoscopically, as each view is processed independently without ensuring stereo coherence. Object-based techniques, on the other hand, apply styles in the 3D world, and project them twice to produce the stereo views. As projection is stereo-coherent, object-based techniques often are as well.

Inconsistencies

I conducted a case study designed to experimentally evaluate the importance of stereo coherence in non-photorealistic rendering. As mentioned in the introduction, stereoscopic perception is best analysed using a perceptual study, in which stimuli are shown to participants who are then asked to comment using introspection. The study compares stereoscopic animations created using two watercolour rendering techniques, as this rendering style shows sufficient visual complexity to exhibit any potential issues.

Case study
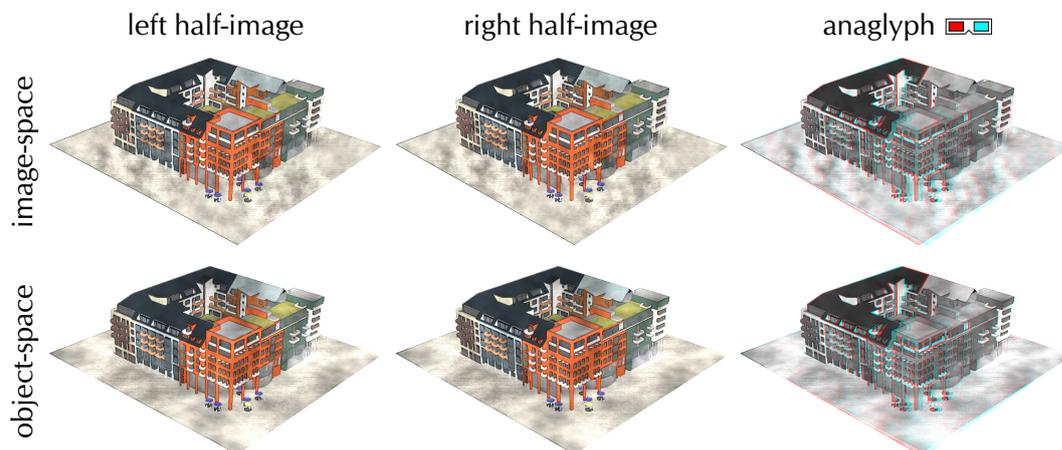
Figure 6.1: Example stimuli shown for the case study on watercolour rendering. The red-cyan anaglyph images on the right are shown without colours for improved viewing comfort. They are also best viewed at a larger zoom level.

6.1.1. Experiment

Stimuli The case study compares animations created using two rendering styles similar to the watercolour rendering technique by Bousseau et al. (2006). Their main difference is how the watercolour's *turbulence flow* texture – which models granulation, or the deposition of ink pigments on paper (Curtis et al., 1997) – is generated. The first method uses the first of the two techniques proposed by Bousseau et al., which creates a dynamic image-space noise texture by blending between six *dynamic canvases* (Cunzi et al., 2003), attached to 3D points in the scene. The second method uses object-space noise created using a dynamic solid texture (Bénard et al., 2009).

Setup Both animations were rendered stereoscopically using quad-buffered OpenGL, displayed using a stereoscopic projector by Lightspeed Design Inc. and viewed through passive, circularly-polarised glasses. Stereo settings, such as screen size and distance, were calibrated before the study, to ensure optimal stereo viewing conditions. Around 5–10 per cent of the population are considered to be 'stereoblind' as they cannot fuse stereo images (Lambooij et al., 2009; Richards, 1970). Hence, each session started by showing a test image with a recessed square on a textured background, similar to a Julesz figure (1964, Figure 2.10), to identify if individual participants were affected by binocular vision problems.

Procedure Each participant was shown rendered versions of the same model of a building, performing a pre-recorded rotation in the centre of the projection screen (for an example see Figure 6.1). The sequences were shown alternately and repeated once more, for about ten seconds each. The participants were then asked how comfortable they found viewing each sequence, and if they could see any differences between them. Finally, they were asked to express a preference for one of the two sequences.

Results The pilot study comprised six participants, all passing the initial stereo test. Five out of the six participants saw differences between the two rendering techniques, and all of those preferred the object-based technique. The participants found that "buildings looked more textured" compared to the image-based technique which they found to have "flatter textures" which were "swimming". An illustration of the mentioned noise behaviour is shown in Figure 6.2.

Conclusions The participants of the case study in watercolour rendering expressed a strong preference for the object-space technique which exhibits better coherence. It also appears to be the case that stereo coherence is closely related to temporal coherence, because object-based techniques are generally more temporally coherent than image-based techniques. Overall, stereo coherence appears to be beneficial in stereoscopic non-photorealistic rendering techniques. The following sections measure stereo coherence objectively and show that it is a good indicator for stereoscopic viewing comfort.
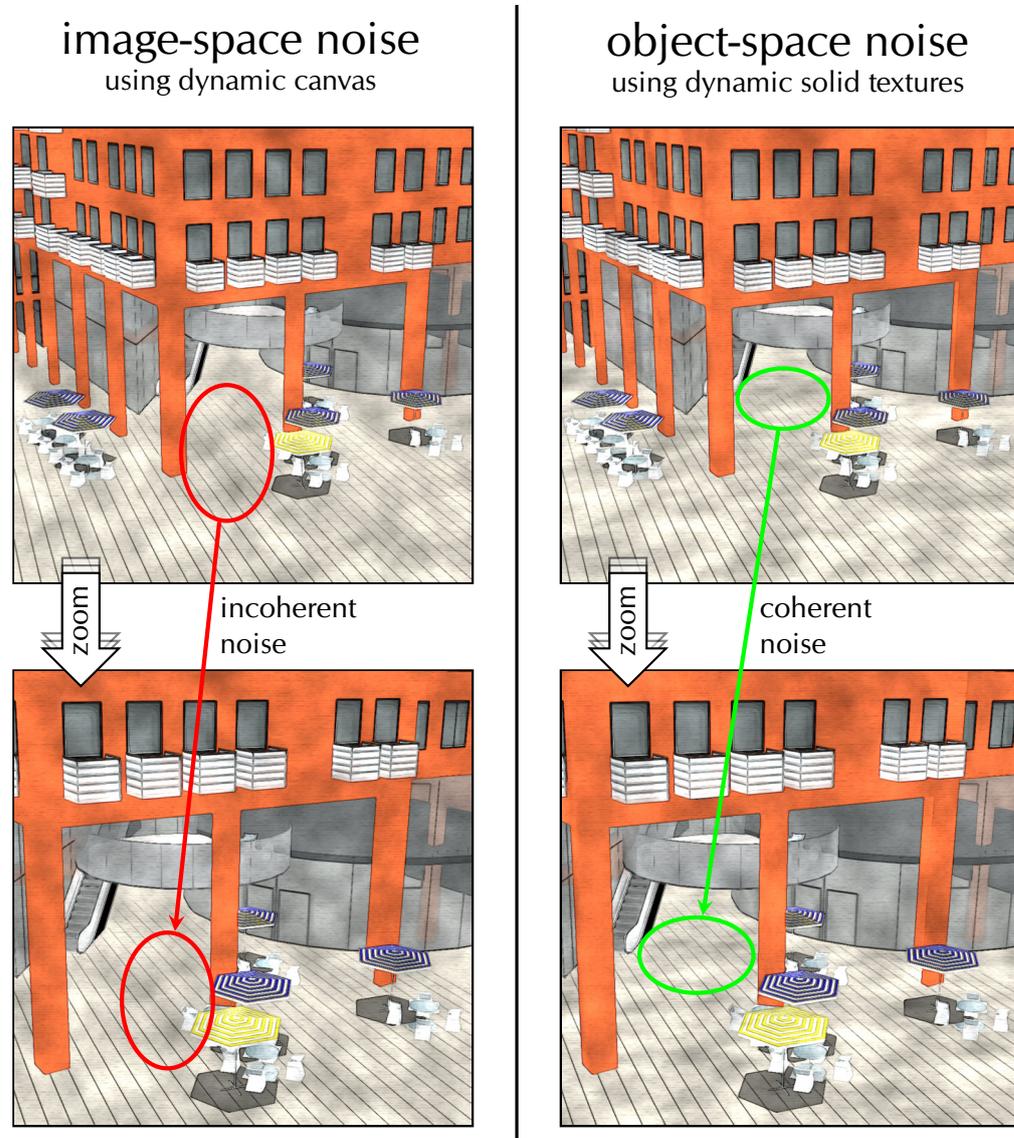
Figure 6.2: Visual comparison of noise coherence when zooming into the scene. **Left:** the motion of the image-space noise is inconsistent with the geometry's motion, resulting in the 'shower door effect'. This is even more apparent when the zoom is animated continuously, or if shown stereoscopically. **Right:** the object-space noise is fixed to the geometry and moves coherently.

6.2. Related work on stereoscopic viewing comfort

Introduction Numerous sources contribute to fatigue and discomfort when viewing stereoscopic content. The first half of this section outlines the different sources of discomfort, and the second half concentrates on work relevant to the study of inconsistencies between the two stereo half-images.

6.2.1. Sources of discomfort

Technological progress Stereoscopic display technology has seen considerable technological progress in the last five years, resulting in increased practicality as well as financial viability. The starkly increasing demand for this technology has been primarily driven by the film industry, but computer games and TV broadcasting also increasingly feature stereoscopic '3D' content.

Discomfort However, even the latest stereoscopic displays can lead to visual discomfort and fatigue (Lambooij et al., 2009; Howarth, 2011) for a variety of reasons related to the physiology of vision, the particular display technology used or the image content.

Variance of inter-pupillary distance Stereoscopic imagery implicitly assumes some particular inter-pupillary distance, which may not match that of the viewer, as no single distance will be right for all people (Dodgson, 2004).

Vergence–accommodation conflicts Most displays require the observer to focus their eyes on the screen instead of the virtual point in 3D space, resulting in the disconnect between vergence and accommodation (Hoffman et al., 2008).

Incorrect depth cues Many depth cues contribute to depth perception (Section 2.2), but some may be inconsistent, such as 'window violations' caused by contradicting occlusion and disparity (Seymour and Neuman, 2011), and other cues may be absent, such as defocus blur away from the plane of fixation (Hoffman et al., 2008).

Absence of motion parallax Moving the head in the real world changes both views due to parallax, but most displays show the same static images (Howarth, 2011).

Crosstalk or ghosting If the stereoscopic views are optically not perfectly separated, each view 'leaks' into the other, creating 'ghost images' (Seuntiëns et al., 2005).

Excessive screen disparity Extreme disparities – when objects are pulled too far out of the screen or pushed back 'beyond infinity' – can lead to the breakdown of binocular fusion and to diplopia, or double vision (Lambooij et al., 2009).

Image discrepancies Image processing operations such as blur, vertical shifts, image compression as well as non-photorealistic filters can lead to irreconcilable differences in the stereo half-images (Kooi and Toet, 2004; Benoit et al., 2008).

The remainder of this chapter concentrates on this last category.

6.2.2. Relevant related work

Kooi and Toet (2004) were the first to analyse experimentally the effect that different image manipulations have on stereoscopic viewing comfort. They considered 35 different image manipulations, including spatial distortions, crosstalk, blur, luminance and contrast adjustments. Based on their experimental results, they concluded that vertical disparity, crosstalk and blur were the factors that most strongly determined visual discomfort. Their exploration of the space of image manipulations informs the design of systems that predict the viewing comfort of stereoscopic display systems. The next Section proposes such a system.

Image discrepancies

Benoit et al. (2008) propose a stereoscopic image quality assessment metric for assessing the impact of compression algorithms like JPEG and JPEG 2000 on the stereoscopic viewing experience. In their metric, they combine conventional image similarity metrics (SSIM, C4) applied to each half-image with a disparity distortion measure, which encodes the difference between the disparity maps of the original and distorted stereo images. Like all image quality metrics, their work relies on the original stereo image being available, while this chapter's model does not.

Stereo image quality metric

Benoit et al.'s disparity maps are computed using state-of-the-art stereo matching techniques like belief propagation and graph cuts (Section 2.4). The computational model in this chapter, on the other hand, is based on normalised cross-correlation – a much simpler stereo matching technique. However, it has been experimentally shown to have similar stereo performance to humans in a number of psychovisual experiments (Banks et al., 2004; Filippini and Banks, 2009; Vlaskamp et al., 2009).

Visual perception

Stavrakis and Gelautz (2005b) describe the inherent problems in creating stereoscopic artwork and outline strategies for extending 'monocular' non-photorealistic rendering techniques to stereoscopic 3D. Their insights draw on their extensive work on stereoscopic painterly rendering, stylisation and sketching (see Section 2.1.3). This chapter expands on two of their criteria: Section 6.3 provides a quantitative measure of 'consistency' by computing stereo correspondences, and Section 6.6.3 localises regions contaminated by 'randomness'.

Stereoscopic artwork

In concurrent work, Didyk et al. (2011) introduce a perceptual model of disparity to estimate the perceived effect of disparity distortions and enhancements. Their model builds on a series of psycho-visual experiments to quantify the effect of changes in disparity magnitude and frequency. While their model can predict which disparity changes are noticeable by human observers, it was not designed to assess viewing comfort of stereoscopic imagery and thus cannot perform this task.

Disparity model

6.3. Computational model of stereo coherence

Introduction I have developed a computational model for objectively estimating the level of visual comfort from a given stereoscopic image. The model combines visual perception research with tools from stereo computer vision to construct a metric for visual comfort based on stereo coherence, that is the consistency of the two half-images.

Interface The input to the computational model is a stereo image and the range of disparities used. The percentage of consistent pixels is output as a coherence score, which was found to be a good indicator for visual comfort (see Section 6.4).

Components The computational model builds on a model of human stereopsis – depth perception from binocular disparity (Section 2.2.2) – recently proposed by Filippini and Banks (2009). As per their model, the left and right stereo half-images are first blurred according to the optical properties of the human eye. Disparity maps for the two half-images are then computed using their local cross-correlator, and then the left-right consistency check (Egnal and Wildes, 2002) is applied to check if corresponding pixels in both disparity maps have consistent disparities.

Results and validation Exemplary results are shown in Section 6.3.4 and the full experimental validation of the model is in Section 6.4. Section 6.6 further describes a set of computational tools extending the model to identify and localise stereo coherence issues.

6.3.1. Optical blur of the human eye

Eye blur The first step in the computational model is a preprocess that applies a mixture of two isotropic 2D Gaussian blurs to the stereo half-images, in order to emulate the optical properties of the human eye. Specifically, this applies the point-spread function of the well-focused eye with a 3 mm pupil after Geisler and Davila (1985):

$$h(x,y) = a \cdot g_{s_1}(x,y) + (1-a) \cdot g_{s_2}(x,y), \tag{6.1}$$

where the 2D Gaussian blur of standard deviation s is

$$g_s(x,y) = \left(2\pi s^2\right)^{-1} \cdot e^{-(x^2+y^2)/2s^2}, \tag{6.2}$$

with $a = 0.583$, $s_1 = 0.433$ arcmin, and $s_2 = 2.04$ arcmin. These parameters assume that the distance between pixels, and therefore the pixel size, is 0.6 arcmin, which roughly corresponds to the spacing between foveal cones.

6.3.2. Local cross-correlator

Normalised cross-correlation Banks et al. (2004) first introduced local cross-correlation as a computational model to help explain why spatial stereoresolution is lower than luminance resolution. The technique they use is known as normalised cross-correlation in computer vision and is described in Section 2.4.1. Recall that it calculates the correlation between

windows of pixels in both stereo half-images. The correlation between windows of disparity d centred on a pixel $\mathbf{p}$ is given by Equation 2.6, that is

$$C_{\text{ZNCC}}(\mathbf{p}, d) = \frac{\displaystyle\sum_{\mathbf{q} \in N_{\mathbf{p}}} (L_{\mathbf{q}} - \overline{L_{\mathbf{p}}}) \cdot (R_{\overline{\mathbf{q}}} - \overline{R_{\overline{\mathbf{p}}}})}{\sqrt{\displaystyle\sum_{\mathbf{q} \in N_{\mathbf{p}}} (L_{\mathbf{q}} - \overline{L_{\mathbf{p}}})^2 \cdot \sum_{\mathbf{q} \in N_{\mathbf{p}}} (R_{\overline{\mathbf{q}}} - \overline{R_{\overline{\mathbf{p}}}})^2}}, \qquad (6.3)$$

where $\overline{I_{\mathbf{p}}}$ is the mean intensity in a square window $N_{\mathbf{p}}$ centred on $\mathbf{p}$ in image I.

In Banks et al.'s original version (2004), the correlation window $N_{\mathbf{p}}$ is a square, but Filippini and Banks recently (2009) proposed to mimic the envelopes associated with cortical receptive fields using a Gaussian weighting function (truncated at 3σ): *Window shape*

$$N_{\mathbf{p}} = e^{-\left(x^2 + y^2\right) / 2\sigma^2}. \qquad (6.4)$$

Filippini and Banks report the best results for $\sigma = 3$ arcmin, which corresponds to 5 pixels, as per the assumptions in the previous section.

The disparity maps are then computed using the winner-take-all technique (as per Section 2.4.3), by selecting the disparity with the highest correlation score. *Winner-take-all*

6.3.3. Left-right check

The left-right consistency check (LRC) is a popular technique in stereo matching for identifying occluded or otherwise inconsistent pixels in disparity maps (Egnal and Wildes, 2002). As described in Section 2.4.4, the check works on two disparity maps: the left-to-right disparity map $d_L(\mathbf{p})$ and the right-to-left disparity map $d_R(\mathbf{p})$. Since both disparity maps are for corresponding views, they should be 'inverses' and disparities of corresponding pixels should sum to zero. A pixel $d_L(\mathbf{p})$ in the left-to-right disparity map is hence considered consistent if this sum falls below a threshold T_{LRC} (which is set to $T_{\text{LRC}} = 1$ throughout this chapter, Equation 2.12):

$$\left| \, d_L(\mathbf{p}) + d_R(\overline{\mathbf{p}}) \, \right| < T_{\text{LRC}}. \qquad (6.5)$$

6.3.4. Example results

Figure 6.3 shows results of the computational model when applied to the computer-generated *City* stereo image as well as five versions with Photoshop filters applied to both half-images independently. As the stereo images get increasingly incoherent, the coherence score computed by the model decreases steadily. It is interesting to note that the original image does not achieve a perfect coherence score according to the model, which is caused by errors in the stereo matching as well as occlusions in the half-images. The next section (6.4) provides a full experimental validation of the model using subjective human ratings of visual comfort.
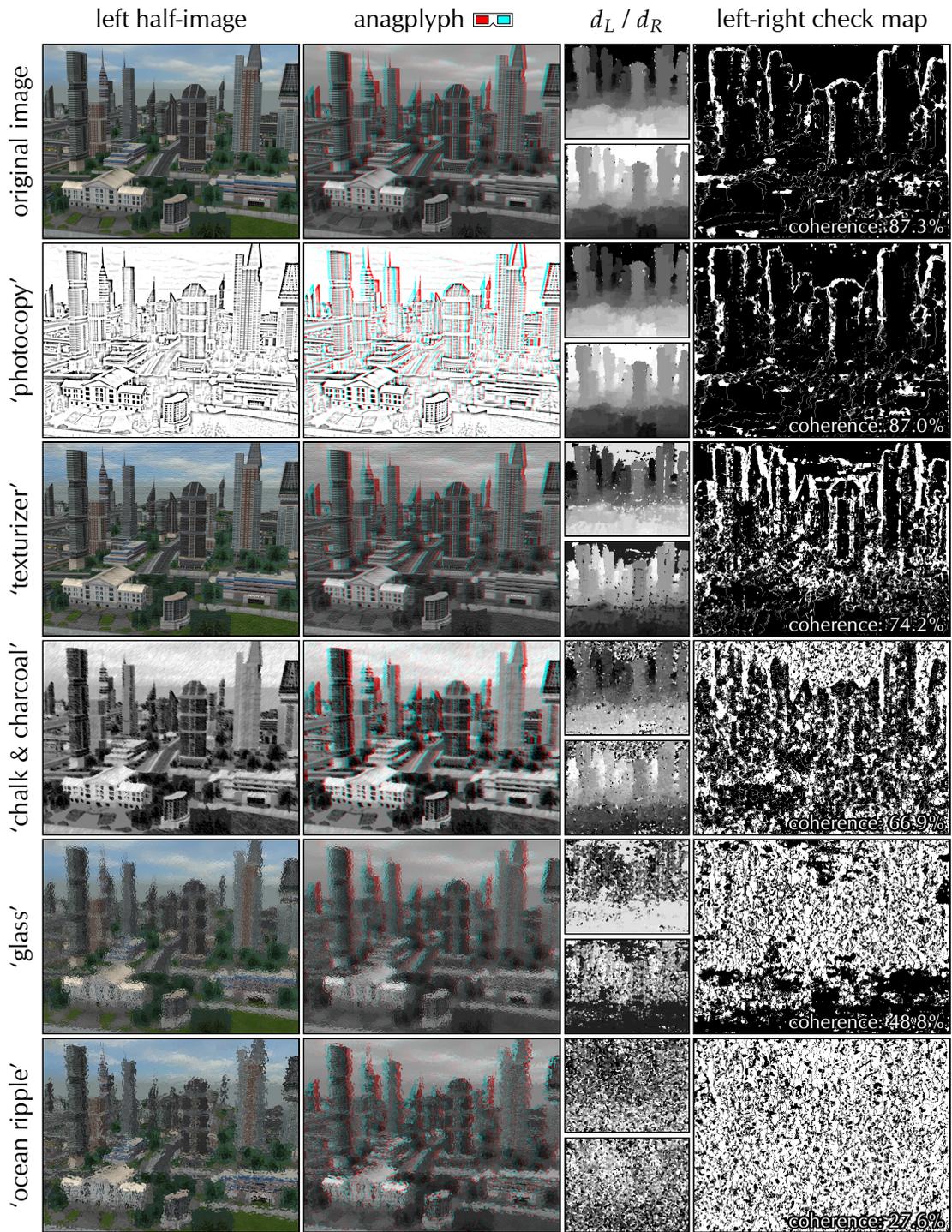
Figure 6.3: Results of the computational model on the *City* stereo image and five versions manipulated using Photoshop filters. The red-cyan anaglyph images may appear less comfortable than on a polarised stereo projector. Note how the disparity maps d_L and d_R, which are scaled to the visible range, are near-inverses of each other. As filters produce less coherent results – from top to bottom – the disparity maps get noisier and more inconsistent as measured using the left-right check. The coherence score in the last column is the percentage of consistent (black) pixels in the left-right check map. If viewed on a display, please zoom in for more visual detail.

6.4. Perceptual study on stereoscopic viewing comfort

To validate the computational model, we[22] conducted a perceptual study in which we asked volunteers to rate the viewing comfort of 80 stereo images. The hypothesis was that there would be a strong correlation between human comfort ratings and the coherence score produced by the model, which would indicate that it would be able to automatically assess the visual comfort levels of stereoscopic images with performance similar to that of human observers. This hypothesis was confirmed by the results of the experiment, as discussed in Sections 6.4.4 and 6.4.5.

Motivation and hypothesis

6.4.1. Experimental setup

We recruited 20 participants (12 male, 8 female) between the ages of 20 and 60; all had normal or corrected-to-normal vision and stereopsis (self-reported).

Participants

Figure 6.4 shows our experimental setup. Stereoscopic 3D images were displayed using a DepthQ projector by Lightspeed Design Inc. and viewed through passive, circularly-polarised glasses from two chairs placed side-by-side in the centre of the room. The projected stimuli did not show any visible crosstalk.

Setup
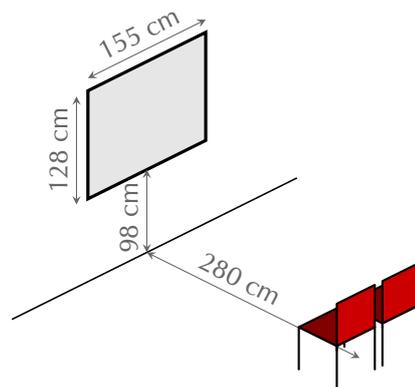
Figure 6.4: Experimental setup for the perceptual study.

6.4.2. Stimuli

In a pilot study, we considered six stereoscopic images and 51 Photoshop filters. We created three of the images and used three from the Middlebury stereo datasets (Scharstein and Szeliski, 2003). To design an experiment that could be completed in reasonable time, we selected a subset[23] of the 306 image–filter combinations. We chose four images (Figure 6.5) and 19 filters (Table 6.1), so including the four original images, a total of 80 images were shown to be rated by the participants.
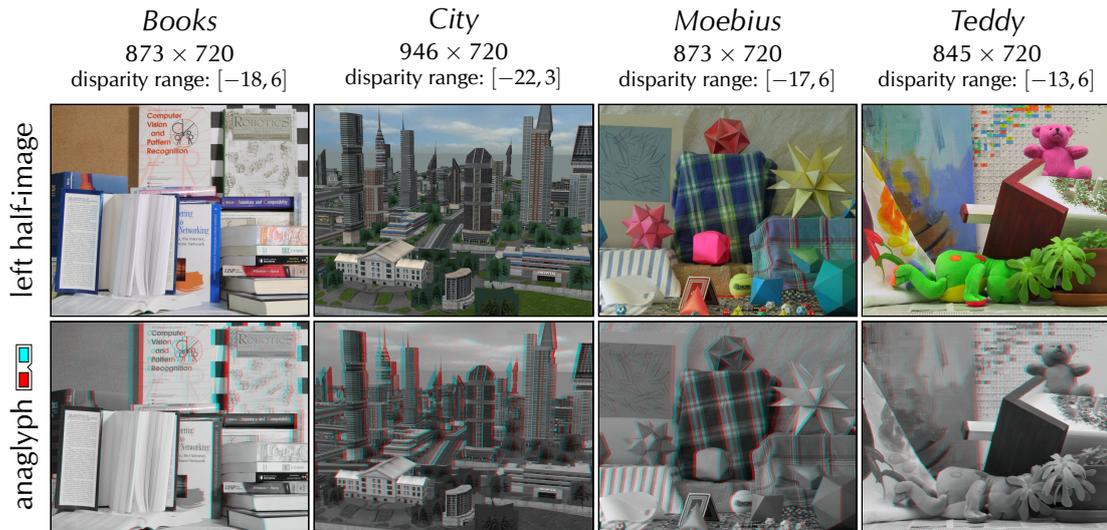
Selected images and filters

The images were rescaled to the projector's height of 720 pixels, and the horizontal disparity was adjusted by shifting the images, so that the front-most object was just in front of the screen, to ease viewing by avoiding excessive disparities.

Preparation of stereo images

---

[22] The perceptual study was jointly designed and carried out with Ian Davies and Lech Świrski (both at the University of Cambridge Computer Laboratory).

[23] We selected the subset so that it covered the full range of artefacts described in Section 6.5.

Books	City	Moebius	Teddy
873×720	946×720	873×720	845×720
disparity range: $[-18, 6]$	disparity range: $[-22, 3]$	disparity range: $[-17, 6]$	disparity range: $[-13, 6]$

Figure 6.5: The four original stereo images used in the perceptual study: *Book*, *Moebius* and *Teddy* from the Middlebury datasets (Scharstein and Szeliski, 2003), and the *City* image created by me.

Chalk & Charcoal	Glass	Palette Knife	Spatter
Craquelure	Grain	Photocopy	Stained Glass
Cutout	Halftone Pattern	Poster Edges	Stamp
Diffuse Glow	Ocean Ripple	Reticulation	Texturizer
Find Edges	Paint Daubs	Rough Pastels	

Table 6.1: The 19 Photoshop filters used in the perceptual study.

6.4.3. Procedure

Rating scheme We asked participants to rate each of the 80 images for viewing comfort on a five-point Likert scale ranging from 1 (very uncomfortable) to 5 (completely comfortable) and we clearly explained that they should rate physical comfort rather than aesthetic quality of the images.

Method We scheduled participants to complete the experiment in pairs to save time; no discussion between them was allowed. For each pair, the order of images was randomised for counter-balancing. Before beginning the main experiment, we showed ten additional images to allow the participants to familiarise themselves with the experimental environment as well as the range of comfort ratings. For each image, the participants were given as long as they wanted to rate it and they were allowed to change their minds until they were content with each rating.

Data collection The ratings were collected using two laptops which transmitted each participant's rating to a third computer, which then wrote the ratings to disk. The images were manually advanced shortly after the last change of rating. Each experimental run took around 15 minutes to complete.

6.4.4. Analysis of correlation

The perceptual study produced 1600 human comfort ratings, as 20 participants rated each of the 80 stereo images that were shown. The computational model also produces a rating for each of the images, albeit a coherence score in the range from 0 to 100 per cent, whereas the human ratings are given on a five-point Likert scale from 1 to 5. Upon visual inspection, scores and ratings appeared to be linked, with a largely linear relationship (Figure 6.6).

Human ratings & coherence scores

This relationship between human comfort ratings and the model can be analysed using correlation coefficients, specifically the correlation between 80-element vectors representing a set of comfort ratings. The familiar Pearson correlation coefficient is best suited for analysing linear effects. Table 6.2 summarises the distribution of correlation coefficients between participants, the average comfort rating for each image, as well as the computational model.

Analysis using correlation
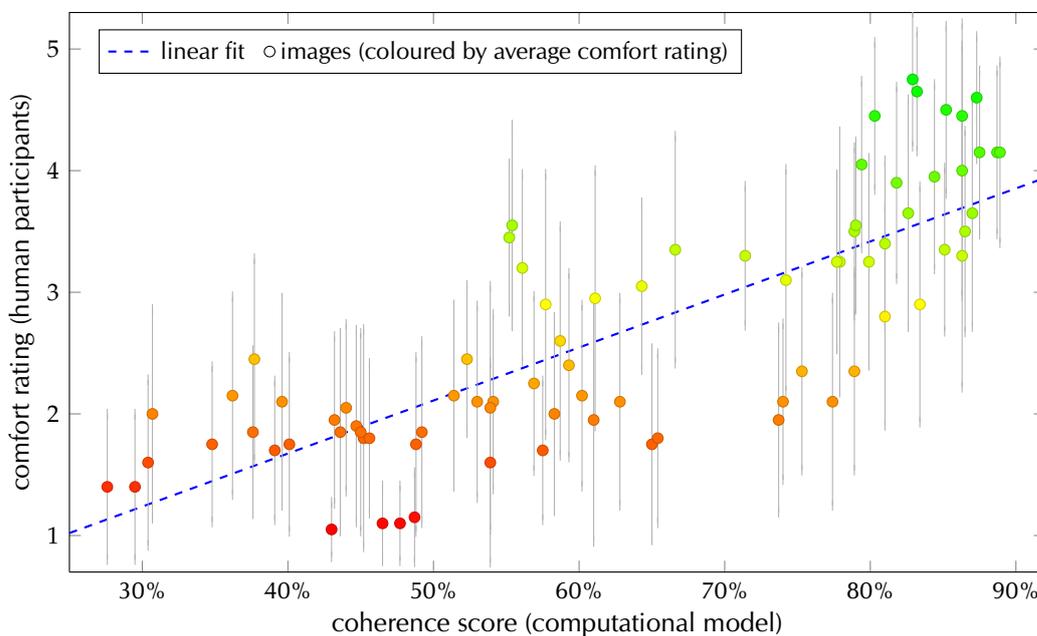
Figure 6.6: Scatter plot of coherence score versus mean human comfort rating for all 80 shown images. The error bars indicate one standard deviation of uncertainty.

	mean	percentiles				
		min	25%	50%	75%	max
participant – participant	0.69	0.46	0.63	0.70	0.75	0.85
participant – model	0.67	0.46	0.66	0.68	0.71	0.79
mean – participant	0.84	0.70	0.81	0.85	0.89	0.91
mean – model	0.80	—	—	—	—	—

Table 6.2: Distribution of Pearson correlation coefficients between participants and other participants (row 1) as well as the computational model (row 2), and between the mean comfort rating and participants (row 3) as well as the computational model (row 4). The bottom row only shows a single correlation coefficient, and not a distribution of them.

Pairwise correlation The upper half of Table 6.2 shows that the correlation between the model and any participant is very similar to the correlation between pairs of participants. The average correlation between pairs is also similar. The conclusion is that the model is as indicative of a particular participant's viewing comfort as any other participant's comfort ratings. In other words, the model is 'as good' as any other participant.

Correlation to mean comfort rating However, comfort ratings vary between participants, as evidenced by the relatively low correlation between pairs of participants ($r = 0.67$). For this reason, the lower half of Table 6.2 compares ratings to the mean comfort rating, which is the average rating vector of all 20 participants. The model outperforms the lowest quartile of participants in terms of correlation with the mean comfort rating. Overall, the computational model is strongly correlated with the mean comfort rating ($r = 0.80$, $p = 6.6 \times 10^{-19}$). The coefficient of determination is $R^2 = r^2 = 0.64$. A linear model is thus a good fit as 64 per cent of the variance in the mean human comfort ratings can thus be explained by the computational model's coherence scores.

6.4.5. Analysis of differences

Predicted comfort To produce a *predicted comfort rating* on the same scale as the human comfort ratings, the model's coherence scores are rescaled linearly using a least squares fit:

$$x' = 4.36 \cdot x - 0.07, \tag{6.6}$$

which achieves a root mean square error (RMSE) of 0.59. The absolute term is close to zero, which suggests a direct relationship between stereo coherence and visual comfort. It also suggests a limited influence of other sources of discomfort (Section 6.2.1) – at least in the context of this perceptual study.

Histogram of differences The remaining differences in the mean and predicted comfort ratings are shown as a histogram in Figure 6.7. The differences are approximately normally distributed, so the linear model is a good fit. The model predicts 61 per cent of images (49 out of 80) to within half a unit of comfort. The baseline against which to compare this is 24 per cent for uniformly random scores in the continuous range $[1, 5]$, and 36 per cent for a constant score of 2.35 (the median of all 80 mean comfort ratings).
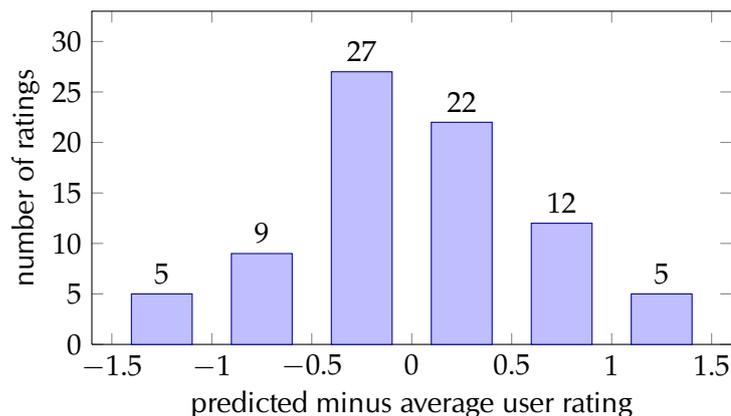
Figure 6.7: The histogram of differences between the ratings predicted by the computational model and the average user rating (the model overestimates visual comfort for positive differences).

The images near the negative end of the axis, where the model underestimates *Negative outliers* visual comfort by more than one unit, are caused by noise, which the human visual system is good at filtering out. A further limitation of the model is that it cannot produce a perfect correlation score, as mentioned in Section 6.3.4.

The outliers where the model overestimates visual comfort are mostly caused by *Positive outliers* the 'shower door effect' (see Section 6.5.2), which the computational model appears to tolerate better than human observers.

6.4.6. Discussion

The experimental setup had a physical pixel size of 2.2 arcmin compared to the *Size of a pixel* 0.6 arcmin assumed in the model. Despite the inconsistency, using the model with this 'incorrect' pixel size results in stronger correlation to the mean comfort rating (0.80 vs 0.71). This is likely due to reduced noise in the disparity maps, as receptive fields are $13\times$ the area compared to using a pixel size of 2.2 arcmin in the model.

The computational model is a good predictor of stereoscopic viewing comfort, as it *Summary* correlates strongly with comfort ratings given by human observers and 61 per cent of predicted comfort ratings are within half a unit of the mean comfort rating.

6.5. Taxonomy of stereo coherence issues

Motivation The model presented in Section 6.3 can evaluate the impact of image manipulations on stereoscopic viewing comfort from a processed stereo image alone. However, it is also important to understand which categories of image manipulations are most detrimental to visual comfort – primarily to avoid them.

Categories of stereo coherence issues Before the perceptual study, Ian Davies and I observed all combinations of six stereo images and 51 Photoshop filters – a total of 306 images. During this process, we took free-form notes on all images and later independently categorised the perceived issues into groups. The categories were nearly identical. We combined them into a taxonomy that we believe represents the major stereo coherence issues in this dataset: binocular rivalry, the shower door effect, and randomness. These categories, which can overlap to some degree, are illustrated by example in Figure 6.8. They are defined and described in more detail in the following sections.

Binocular rivalry   Shower door effect   Randomness
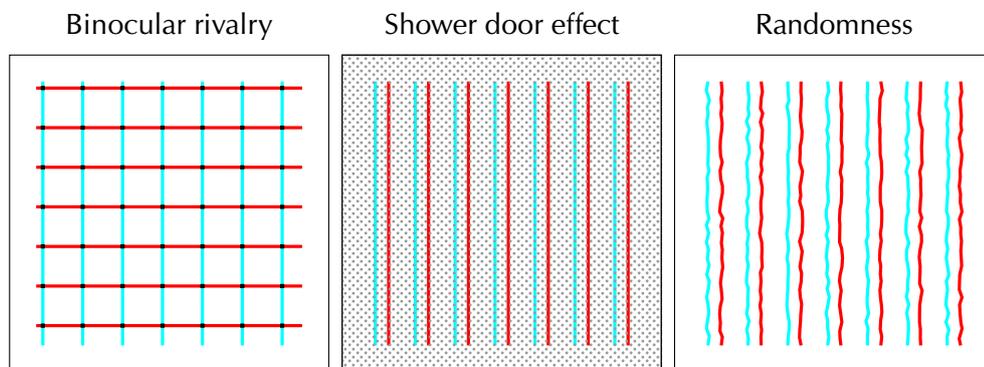
Figure 6.8: Red-cyan anaglyph ■,▫ examples of the identified stereo coherence issues.

6.5.1. Binocular rivalry

Definition Also known as retinal rivalry, binocular rivalry describes alternations in perception that are experienced when mismatched stimuli are presented to the two eyes (Blake, 2001; Blake and Logothetis, 2002). The first detailed description of this phenomenon is due to Wheatstone (1838), who mounted different letters in his stereoscope and described his observations. What he saw was that one eye's stimulus dominates over the other for a few seconds, until the image "breaks into fragments, while fragments of the letter which is about to appear mingle with them, and are immediately after replaced by the entire letter", and that this process would repeat every few seconds.

Sources Binocular rivalry is caused whenever image regions are strongly conflicting between the two stereo half-images, that is if they are not in correspondence with a region in the other half-image. This definition has some overlap with our third category, randomness, so we restrict our definition to regions covering at least a few degrees of visual arc. Binocular rivalry is most prominently caused by Photoshop filters that use morphological operators, segmentation or colour quantisation, as they can remove objects or modify object boundaries. For an example, see Figure 6.9.

6.5.2. Shower door effect

The *shower door effect* is a term commonly used in non-photorealistic rendering (Section 2.1) to describe a look that resembles textured glass in front of the main content of an image. This effect is most easily achieved by compositing identical textures into both stereo half-images, like in the 'texturizer' filter shown in Figure 6.3. The resulting flat, transparent texture has a disparity of zero, which places it exactly at the depth of the screen. *Definition*

If the texture is in front of other objects, it creates visible artefacts in the plane of the screen which can partially obscure or distort what is behind (Akerstrom and Todd, 1988). If it is behind other objects, the situation is worse because there are conflicting depth cues: the 'shower door' appears to be in front of the objects visually but behind them in terms of depth. This conflict increases visual discomfort. *Placement relative to the screen*

A perhaps more deserving example is the 'glass' effect in Figure 6.3, which applies an identical distortion to both half-images. The result looks like a shower door with rippled glass, or, "like looking through a window", as one participant put it. *Canonical example*

6.5.3. Randomness

The final category captures everything where randomness is in play. In general, if the same effect applied twice to the same image produces two noticeably different resulting images, then the stereo half-images are also most likely incoherent. The simplest example is per-pixel noise, such as film grain. *Definition*

While small amounts of noise are tolerated by the human visual system, stronger noise can make it hard to fuse the stereo half-images, which causes visual discomfort. Another example is the 'chalk & charcoal' effect in Figure 6.3, which places strokes randomly and with random length. *Tolerance*

One way to ameliorate the effects of randomness is to fix the seed value of the random number generator. In general, this reduces effects based on randomness to the shower door effect of the previous section, as the same manipulation is now applied to both half-images. One example of this is the 'ocean ripple' filter in Figure 6.3, which has a mean comfort rating of 1.6 ($\sigma = 0.7$). Fixing the seed of the random number generator used by this filter would produce results resembling the 'glass' filter, which has a slightly higher mean comfort rating of 1.9 ($\sigma = 0.7$). *Connection to the shower door effect*

## 6.6. Computational tools for stereo coherence analysis

Introduction & motivation

The computational model described in Section 6.3 is based on the general approach of checking disparity maps for consistency. This allows the model to objectively quantify the degree of stereoscopic coherence, and thus estimate viewing comfort. However, because of this generality, the model cannot differentiate between different types of incoherencies, which is of interest to stereoscopic content creators such as artists. For this purpose, this section presents extensions to the computational model that help to identify and localise the different stereo coherence issues.

6.6.1. Binocular rivalry

Blur left-right check map

The computational model's left-right consistency map highlights inconsistencies between the half-images directly. The influence of occlusion artefacts can be reduced using a Gaussian blur with the same parameters used in windowing the local cross-correlator of Section 6.3.2. This is because the human visual system does not operate on a 'per-pixel' level, but rather using larger receptive fields. Some results of this approach are shown in Figure 6.9.
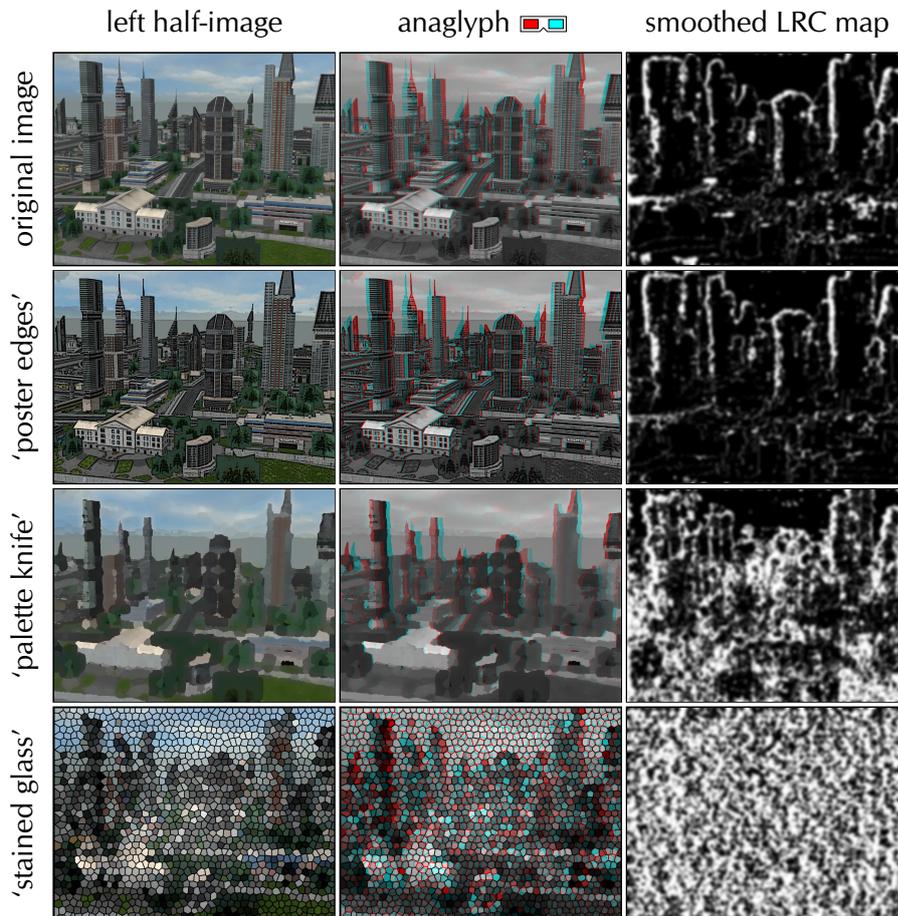
Figure 6.9: Examples of binocular rivalry detection: 'poster edges' cause hardly any binocular rivalry, 'palette knife' shows artefacts near object boundaries, and 'stained glass' is globally incoherent.

6.6.2. Shower door effect

The coherence analysis detects stereo inconsistencies, but the shower door effect *Different approach*
does not cause inconsistencies. Nevertheless, it causes the 3D structure perceived
due to stereopsis to be ambiguous or incorrect (Akerstrom and Todd, 1988). The
normalised cross-correlator (Section 6.3.2) can be modified to detect these issues.

The cross-correlator calculates each pixel's most likely disparity as the one which *Likely disparities*
has globally maximal correlation over all disparities. In the shower door effect,
a texture is blended onto both stereo half-images, with zero disparity. Given the
blending, the texture's disparity may not be the most likely disparity, and would
therefore be discarded in the winner-take-all approach. However, it may still be a
'likely disparity' with locally maximal correlation.

We[24] accumulate the likely disparities of all pixels using a histogram, which *Histogram of likely*
reveals the likelihood of each disparity. With this definition, peaks in the histogram *disparities*
correspond to many pixels with the same likely disparity. A strong peak at zero
disparity is therefore indicative of the shower door effect, as shown in Figure 6.10.
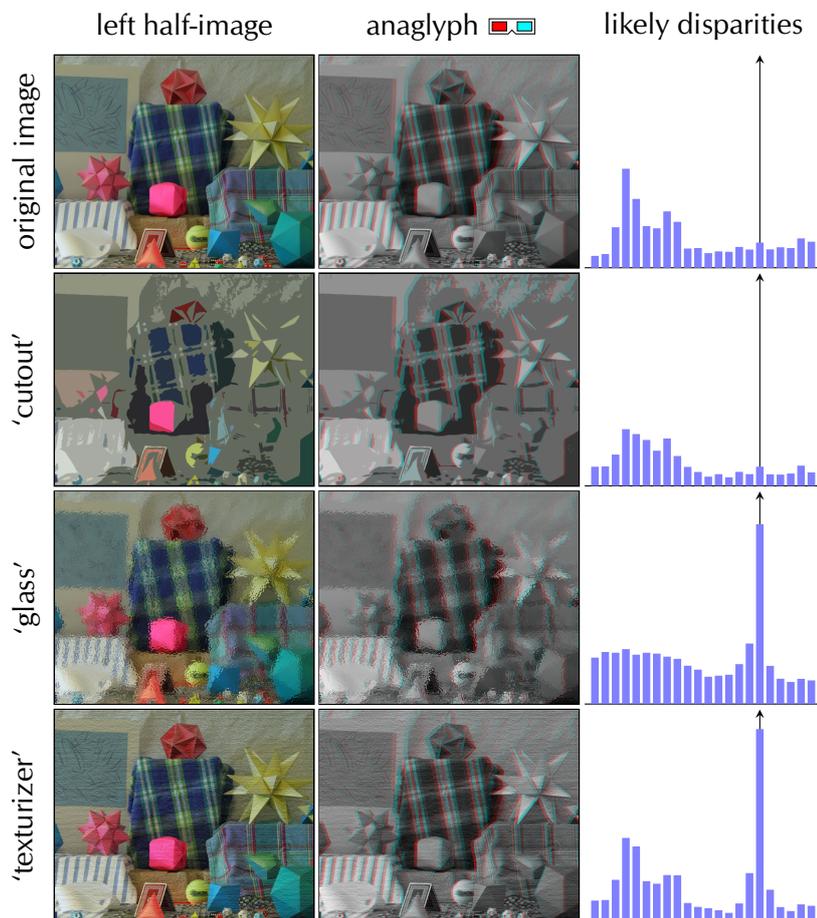
Figure 6.10: Results of the 'shower door effect' detection. Notice that the bottom two effects have large
peaks at a likely disparity of zero, which is indicative of the shower door effect.

---

[24] Joint work with Lech Świrski, who also implemented this technique.

6.6.3. Randomness

Elementary check   As mentioned in Section 6.5.3, an elementary check for randomness is to apply an operation twice to the same image and to compare the resulting images. This comparison could also be automated using image quality metrics such as structural similarity (SSIM; Wang et al., 2004b) or the visible differences predictor (VDP; Daly, 1992). However, it is not always possible to apply an effect twice, for example when working with existing imagery.

Image-based cross-check   A more general approach is the following image-based cross-check, which compares the colours of consistent pixels. Specifically, the corresponding colours of all consistent pixels, as indicated by the left-right check map (Section 6.3.3), are compared using the ΔE_{ab}^* colour difference in the CIELAB colour space. Pixels with inconsistent disparities are set to zero. As before (Section 6.6.1), this is followed by a Gaussian blur to mimic the behaviour of receptive fields.

Interpretation   Figure 6.11 shows some examples of this approach, where the image-based cross-check map is scaled to the range of *just-noticeable differences* from 0 (black, $\Delta E_{ab}^* = 0$) to 2 (white, $\Delta E_{ab}^* = 6$). Images with correctly calculated disparity, but rivalry due to noise, show problem areas as large, mostly white regions in the image cross-check.
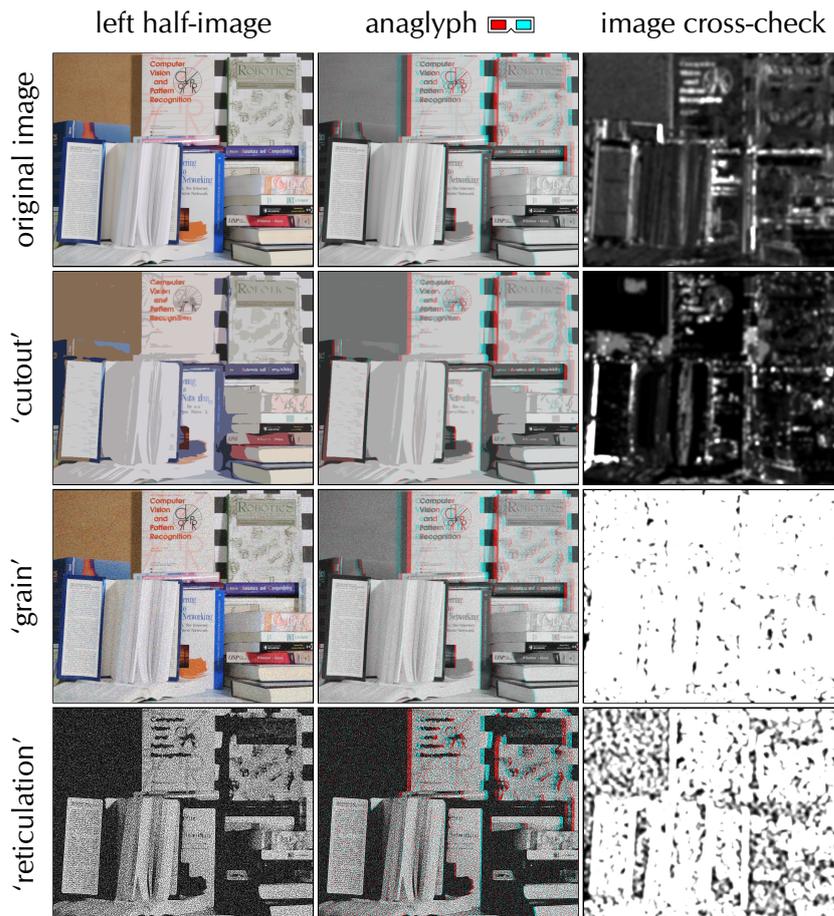
Figure 6.11: Results of the image-based cross-check. The lower two filters show a lot of randomness.

6.7. Conclusion

I have demonstrated the first computational model to estimate the viewing comfort of stereoscopic images. By building on research in visual perception and stereo computer vision, my model measures the coherence of stereo half-images, which I hypothesised to be indicative of viewing comfort. Using a perceptual study with 20 participants, I could demonstrate that this is indeed the case, as the computed coherence scores strongly correlate with human comfort ratings.

Summary

This makes my model ideal for automatically assessing computer-generated stereo-scopic content without the need for perceptual studies. I have also described the primary obstacles to stereo coherence, which are retinal rivalry, the shower door effect and randomness in the two half-images. Furthermore, I proposed extensions of my model to detect and localise these issues in stereo images.

Applications

Future work could improve the performance of the model further, for example by using occlusion handling to allow a more accurate comfort prediction for stereo images that are completely coherent, or using global stereo matching techniques which would improve the handling of weakly textured areas. The model could also be extended to take other sources of discomfort (Section 6.2.1) into account. An interesting and orthogonal avenue of future work is the analysis of the temporal dynamics of stereoscopic viewing comfort, perhaps by quantifying the combined stereo-temporal coherence.

Improving the model

<div style="text-align: right; font-size: 3em;">**7**</div>

Conclusions

In this dissertation, I have considered the full life cycle of RGBZ videos: from video capture via filtering and processing to evaluation of display. On this journey, I have proposed two techniques for capturing RGBZ videos, which are, respectively, based on spatiotemporal stereo matching (Chapter 3), and on a sensor fusion approach combining a time-of-flight camera with a normal colour video camera (Chapter 4). Using the recorded RGBZ videos, I next demonstrated a variety of effects that rely on high-quality depth data, such as video relighting and stereoscopic rendering (Chapter 5). Finally, these stereoscopic renderings inspired me to study the aspects leading to visual discomfort, which resulted in a model for predicting stereoscopic viewing comfort (Chapter 6). *Summary*

The following sections summarise the main contributions in these three areas in more detail before revisiting my hypotheses and discussing some of the potential research directions that are opened up by my work. *Structure*

7.1. Coherent depth acquisition

I have considered two approaches for capturing depth videos to create temporally coherent RGBZ videos in this dissertation: using spatiotemporal stereo matching and time-of-flight sensor fusion. *Approaches*

The spatiotemporal stereo matching approach, described in Chapter 3, starts with the *dual-cross-bilateral grid* – a novel real-time stereo matching technique derived by accelerating a reformulation of an existing cost aggregation approach. This is the basis for a temporal extension that aggregates costs over spatiotemporal support windows. The result is increased temporal coherence of the disparity maps – particularly in the presence of image noise. Five synthetic stereo videos with ground truth disparity maps were created to qualitatively evaluate this technique, which processes these videos at 11 Hz (for 400×300 videos with 64 disparities). *Contributions to stereo matching*

The time-of-flight sensor fusion system presented in Chapter 4 combines data from a noisy, low-resolution time-of-flight camera and a high-resolution colour video camera into a coherent, noise-free RGBZ video. The system consists of a three-step video processing pipeline that aligns the depth and RGB video streams, efficiently removes and fills invalid and noisy geometry, and finally uses a spatiotemporal *Contributions to ToF sensor fusion*

filtering approach to increase the spatial resolution of the depth data and strongly reduce the depth measurement noise. The results are high-quality RGBZ videos, processed at about 5 Hz (for a 584×506 RGBZ video).

First hypothesis Having reviewed the contributions of Chapters 3 and 4, I can confirm the validity of my first hypothesis from Section 1.2, that is that

> H1. It is possible to reconstruct dynamic scene geometry coherently at interactive frame rates.

Both geometry capture approaches confirm this hypothesis independently. However, the time-of-flight sensor fusion approach produces higher-quality RGBZ videos, which enable more advanced RGBZ video processing effects than are possible with the disparity maps computed using the spatiotemporal stereo matching approach.

Improving stereo performance The stereo matching approach of Chapter 3 is one of the first techniques to address the problem of processing stereo videos. This problem is certainly not solved yet. Clearly, there is room for substantial improvements in quality. An obvious starting point is to experiment with different acceleration approaches to the bilateral filter (Section 2.5.3) to enable full-colour filtering of the cost volume which promises to deliver noticeable performance improvements. Rhemann et al. (2011) propose such an approach which provides greatly improved performance while also further reducing run time. I expect further advances in quality can be achieved by lifting the assumption of frontoparallel surfaces by using slanted support windows, and by incorporating explicit occlusion handling. All these aspects would improve the single-frame technique. For videos, adapting the spatiotemporal support windows to the underlying scene motion, similar to the geometry filter in Section 4.3, would most likely reduce artefacts in the disparity map caused by fast motions. For this to work well, high-quality optical flow is required, which could potentially be jointly estimated with the disparity flow – the change of disparities over time.

Evaluating stereo matching on videos More interesting from a research perspective is the open question of how to evaluate stereo correspondence techniques working on stereo videos. Although we have created a set of synthetic videos with ground truth disparity maps that techniques can be compared against, it is not clear how to compute meaningful performance metrics that objectively quantify flickering and temporal noise in disparity videos. In addition, it would be desirable to broaden the range of ground truth videos available to also include live action footage – although it would be difficult to obtain ground truth depth maps. Once these question are settled, it would be prudent to set up an evaluation website in the spirit of Middlebury stereo evaluation website which would allow techniques to be evaluated objectively and ranked accordingly.

Improving ToF sensor fusion The spatiotemporal geometry filter in Section 4.3 results in a fairly strong smoothing effect that is a byproduct of removing all temporal noise. While suppressing noise is the top priority, preserving prominent features in the RGBZ video should come a close second. At least for the moment, however, there appears to be an inherent trade-off between the strength of noise reduction and the preservation of smaller features. Breaking this connection would be an important contribution to research. Independently, the computational complexity of the proposed processing pipeline could likely be reduced – perhaps using a multi-resolution approach or an efficient

bilateral filter approximation. Lastly, the quality of distance maps is also influenced by the quality of the optical flow which tends to be unreliable due to motion blur, large displacements and occlusions. An interesting avenue of future work would therefore be an optical flow formulation that respects depth discontinuities, as this would hopefully prevent 'smearing' artefacts in the filtered distance map.

The proposed RGBZ video capturing approaches still require specialised hardware and considerable processing power to filter the captured data. With miniaturisation of the camera hardware, algorithmic optimisations and improvements in hardware performance, such RGBZ video cameras could soon become available commercially. Some consumer devices exist already, like the *Fujifilm FinePix REAL 3D W series* digital stereo cameras, or the *Microsoft Kinect* sensor. However, these devices still lack the processing power to create aligned, high-quality RGBZ videos, and instead just capture raw image data, which cannot easily be used for RGBZ video effects.

Commodification of RGBZ video cameras

7.2. RGBZ video effects

RGBZ videos provide additional geometric information over normal colour videos. I showed in Chapter 5 that this enables a range of video processing effects that are not achievable from colour videos alone, because the geometric information is a critically important ingredient to create these effects. One example for this is the proposed video relighting technique, which requires high-quality surface normals to produce plausible results. I also presented non-photorealistic rendering techniques that benefit from geometric information to help place lines, shadows and brush strokes, producing clear improvements over existing video-based techniques. Perhaps the strongest application of RGBZ videos is the ability to synthesise stereo videos, which allows other RGBZ video effects to be applied stereoscopically.

Contributions to RGBZ video effects

The range of RGBZ videos effects presented directly supports my second hypothesis from Section 1.2, that is that

Second hypothesis

> H2. RGBZ videos facilitate a variety of advanced video processing and non-photorealistic rendering effects.

The presented effects demonstrate beyond doubt the advantages of RGBZ videos. However, the techniques I described are only a few samples from a much larger space of possible RGBZ video effects that I did not explore in detail. Many existing effects could be extended and enhanced using the available depth or surface normal information, and completely new effects could be created to exploit the combined RGBZ data. I believe these effects provide benefits for both commercial and personal applications. The video segmentation and relighting effects are examples which provide practical improvements for commercial applications in video editing and post-production. Personal users, on the other hand, could use the larger and more expressive arsenal of non-photorealistic rendering techniques to help express the emotions captured in a video, for example to conserve emotional holiday memories.

Commodification of RGBZ video effects

7.3. Stereoscopic viewing comfort

Contributions to stereo comfort Predicting the viewing comfort (or discomfort) of stereoscopic images is the topic of Chapter 6. In this chapter, I described the first computational model that predicts the visual comfort of stereoscopic images, which I validated using a perceptual study. This study concluded that the predicted comfort scores correlate strongly with human comfort ratings, which makes them ideal for automatic comfort assessment – without the need for costly and lengthy perceptual studies. The preparation of the study has furthermore shown that there are three broad categories of stereo coherence issues which affect human viewing comfort, and I have described computational tools to detect and localise these issues.

Third hypothesis The results of the perceptual study are compared to the predicted comfort scores in Section 6.4.4, which concluded that the computational model is 'as good' as any participant in the study, and overall, the model is strongly correlated with the mean comfort rating. This confirms my third, and final, hypothesis of Section 1.2:

> H3. Stereoscopic viewing comfort can be predicted from stereoscopic images alone.

Supporting the stereographer Some options to improve the model of stereoscopic viewing comfort were already discussed in Section 6.7, so I shall not repeat them here. Instead, I will discuss applications beyond image manipulations and computer-generated imagery which are the focus of Chapter 6. One such application is live-action stereoscopic filming, which is usually supported by a stereographer – a member of the film crew who oversees 'all things stereo' and needs to rely on their judgement regarding all settings of the stereo camera rigs. Mistakes such as misaligned cameras, vertical disparity, mismatched zoom, focus, white balance or brightness can be difficult and costly to fix in post-production (Technicolor, 2011). However, these problems could potentially be identified algorithmically – and perhaps even corrected automatically – which would result in 'better stereo' and also save money. Some of this is already implemented in the commercial stereoscopic analyser STAN[25] by Fraunhofer, which also provides basic tools for correcting some of the problems.

Last words However, I believe that there is always room for improvement.

---

[25] http://www.hhi.fraunhofer.de/stan/

Bibliography

Andrew Adams, Natasha Gelfand, Jennifer Dolson, and Marc Levoy. Gaussian KD-trees for fast high-dimensional filtering. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 28(3):21:1–12, August 2009. doi: 10.1145/1531326.1531327. URL http://graphics.stanford.edu/papers/gkdtrees/.

Andrew Adams, Jongmin Baek, and Abe Davis. Fast high-dimensional filtering using the permutohedral lattice. *Computer Graphics Forum (Proceedings of Eurographics)*, 29(2):753–762, May 2010. URL http://graphics.stanford.edu/papers/permutohedral/.

Aseem Agarwala. SnakeToonz: a semi-automatic approach to creating cel animation from video. In *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, pages 139–146, 163, June 2002. doi: 10.1145/508530.508554. URL http://agarwala.org/Pages/snaketoonz.html.

Amit Agrawal. Non-photorealistic rendering: Unleashing the artist's imagination [graphically speaking]. *IEEE Computer Graphics and Applications*, 29(4):81–85, July/August 2009. ISSN 0272-1716. doi: 10.1109/MCG.2009.61.

Robin A. Akerstrom and James T. Todd. The perception of stereoscopic transparency. *Attention, Perception & Psychophysics*, 44(5):421–432, September 1988. ISSN 1943-3921. doi: 10.3758/BF03210426.

Volker Aurich and Jörg Weule. Non-linear Gaussian filters performing edge preserving diffusion. In *Proceedings of DAGM Symposium*, pages 538–545, September 1995. ISBN 3-540-60293-3.

Martin S. Banks, Sergei Gepshtein, and Michael S. Landy. Why is spatial stereoresolution so low? *Journal of Neuroscience*, 24(9):2077–2089, March 2004. doi: 10.1523/jneurosci.3852-02.2004.

Danny Barash. Fundamental relationship between bilateral filtering, adaptive smoothing, and the nonlinear diffusion equation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(6):844–847, June 2002. ISSN 0162-8828. doi: 10.1109/TPAMI.2002.1008390.

Pascal Barla, Joëlle Thollot, and Lee Markosian. X-Toon: an extended toon shader. In *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, pages 127–132, 164, June 2006. doi: 10.1145/1124728.1124749. URL http://maverick.inria.fr/Publications/2006/BTM06a/.

Alberto Bartesaghi, Guillermo Sapiro, Tom Malzbender, and Dan Gelb. Three-dimensional shape rendering from multiple images. *Graphical Models*, 67(4):332–346, July 2005. ISSN 1524-0703. doi: 10.1016/j.gmod.2005.02.002.

Ronen Basri, David Jacobs, and Ira Kemelmacher. Photometric stereo with general, unknown lighting. *International Journal of Computer Vision*, 72(3):239–257, May 2007. doi: 10.1007/s11263-006-8815-7.

Alexandre Benoit, Patrick Le Callet, Patrizio Campisi, and Romain Cousseau. Quality assessment of stereoscopic images. *EURASIP Journal on Image and Video Processing*, 2008:659024:1–13, 2008. doi: 10.1155/2008/659024.

Marcelo Bertalmio, Guillermo Sapiro, Vincent Caselles, and Coloma Ballester. Image inpainting. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 417–424, August 2000. ISBN 1-58113-208-5. doi: 10.1145/344779.344972.

Randolph Blake. A primer on binocular rivalry, including current controversies. *Brain and Mind*, 2(1):5–38, April 2001. ISSN 1389-1987. doi: 10.1023/A:1017925416289.

Randolph Blake and Nikos K. Logothetis. Visual competition. *Nature Reviews Neuroscience*, 3(1):13–21, January 2002. doi: 10.1038/nrn701.

Michael Bleyer and Margrit Gelautz. Graph-based surface reconstruction from stereo pairs using image segmentation. In *Videometrics*, volume 5665 of *Proceedings of SPIE*, page 288, January 2005. doi: 10.1117/12.586502.

Adrien Bousseau, Matt Kaplan, Joëlle Thollot, and François X. Sillion. Interactive watercolor rendering with temporal coherence and abstraction. In *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, pages 141–149, June 2006. doi: 10.1145/1124728.1124751.

Adrien Bousseau, Fabrice Neyret, Joëlle Thollot, and David Salesin. Video watercolorization using bidirectional texture advection. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 26(3):104, July 2007. ISSN 0730-0301. doi: 10.1145/1276377.1276507.

Yuri Boykov, Olga Veksler, and Ramin Zabih. Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239, November 2001. ISSN 0162-8828. doi: 10.1109/34.969114.

David Brewster. *The Stereoscope: Its History, Theory, and Construction*. John Murray, London, 1856. Reprinted 1971 by Morgan & Morgan.

Stephen Brooks. Mixed media painting and portraiture. *IEEE Transactions on Visualization and Computer Graphics*, 13(5):1041–1054, September/October 2007. ISSN 1077-2626. doi: 10.1109/TVCG.2007.1025.

Thomas Brox, Andrés Bruhn, Nils Papenberg, and Joachim Weickert. High accuracy optical flow estimation based on a theory for warping. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 3024 of *Lecture Notes in Computer Science*, pages 25–36, May 2004. doi: 10.1007/978-3-540-24673-2_3.

Pierre Bénard, Adrien Bousseau, and Joëlle Thollot. Dynamic solid textures for real-time coherent stylization. In *Proceedings of the Symposium on Interactive 3D Graphics and Games (I3D)*, pages 121–127, February 2009. doi: 10.1145/1507149.1507169. URL http://maverick.inria.fr/Publications/2009/BBT09/.

Pierre Bénard, Forrester Cole, Aleksey Golovinskiy, and Adam Finkelstein. Self-similar texture for coherent line stylization. In *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, pages 91–97, June 2010. ISBN 978-1-4503-0125-1. doi: 10.1145/1809939.1809950. URL http://maverick.inria.fr/Publications/2010/BCGF10/.

John Canny. A computational approach to edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 8(6):679–698, November 1986. ISSN 0162-8828. doi: 10.1109/TPAMI.1986.4767851.

Derek Chan, Hylke Buisman, Christian Theobalt, and Sebastian Thrun. A noise-aware filter for real-time depth upsampling. In *Proceedings of the ECCV Workshop on Multi-camera and Multi-modal Sensor Fusion Algorithms and Applications*, October 2008.

Jiawen Chen, Sylvain Paris, and Frédo Durand. Real-time edge-aware image processing with the bilateral grid. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 26(3):103:1–9, July 2007. ISSN 0730-0301. doi: 10.1145/1276377.1276506.

Forrester Cole, Aleksey Golovinskiy, Alex Limpaecher, Heather Stoddart Barros, Adam Finkelstein, Thomas Funkhouser, and Szymon Rusinkiewicz. Where do people draw lines? *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 27(3):88:1–11, August 2008. ISSN 0730-0301. doi: 10.1145/1360612.1360687. URL http://gfx.cs.princeton.edu/pubs/Cole_2008_WDP/.

Forrester Cole, Kevin Sanik, Doug DeCarlo, Adam Finkelstein, Thomas Funkhouser, Szymon Rusinkiewicz, and Manish Singh. How well do line drawings depict shape? *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 28(3):28:1–9, August 2009. doi: 10.1145/1531326.1531334. URL http://gfx.cs.princeton.edu/pubs/Cole_2009_HWD/.

John P. Collomosse and Peter M. Hall. Salience-adaptive painterly rendering using genetic search. *International Journal on Artificial Intelligence Tools (IJAIT)*, 15(4):551–576, August 2006. doi: 10.1142/S0218213006002813.

John P. Collomosse and Jan Eric Kyprianidis. Artistic stylization of images and video. In *Eurographics Tutorials*, April 2011. URL http://kahlan.eps.surrey.ac.uk/EG2011/.

John P. Collomosse, David Rowntree, and Peter M. Hall. Stroke surfaces: Temporally coherent artistic animations from video. *IEEE Transactions on Visualization and Computer Graphics*, 11(4):540–549, September/October 2005. ISSN 1077-2626. doi: 10.1109/TVCG.2005.85.

A. Criminisi, P. Perez, and K. Toyama. Region filling and object removal by exemplar-based image inpainting. *IEEE Transactions on Image Processing*, 13(9):1200–1212, September 2004. doi: 10.1109/TIP.2004.833105.

Matthieu Cunzi, Joëlle Thollot, Sylvain Paris, Gilles Debunne, Jean-Dominique Gascuel, and Frédo Durand. Dynamic canvas for non-photorealistic walkthroughs. In *Proceedings of Graphics Interface (GI)*, June 2003. URL http://maverick.inria.fr/Publications/2003/CTPDGD03/.

Cassidy J. Curtis, Sean E. Anderson, Joshua E. Seims, Kurt W. Fleischer, and David H. Salesin. Computer-generated watercolor. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 421–430, August 1997. ISBN 0-89791-896-7. doi: 10.1145/258734.258896.

James E. Cutting and Peter M. Vishton. Perceiving layout and knowing distances: The integration, relative potency, and contextual use of different information about depth. In William Epstein and Sheena Rogers, editors, *Perception of Space and Motion*, volume 5 of *Handbook of Perception and Cognition*, pages 69–117. Academic Press, 1995. ISBN 0122405307.

Scott J. Daly. Visible differences predictor: an algorithm for the assessment of image fidelity. In *Human Vision, Visual Processing, and Digital Display*, volume 1666 of *Proceedings of SPIE*, February 1992. doi: 10.1117/12.135952.

James Davis, Diego Nehab, Ravi Ramamoorthi, and Szymon Rusinkiewicz. Spacetime stereo: a unifying framework for depth from triangulation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 27(2):296–302, February 2005. doi: 10.1109/TPAMI.2005.37.

Doug DeCarlo, Adam Finkelstein, Szymon Rusinkiewicz, and Anthony Santella. Suggestive contours for conveying shape. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 22(3):848–855, July 2003. ISSN 0730-0301. doi: 10.1145/882262.882354.

Doug DeCarlo, Adam Finkelstein, and Szymon Rusinkiewicz. Interactive rendering of suggestive contours with temporal coherence. In *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, pages 15–145, June 2004. ISBN 1-58113-887-3. doi: 10.1145/987657.987661.

Douglas DeCarlo and Anthony Santella. Stylization and abstraction of photographs. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 21(3):769–776, July 2002. doi: 10.1145/566570.566650.

Philippe Decaudin. Cartoon looking rendering of 3D scenes. Research Report 2919, INRIA, June 1996. URL http://www.antisphere.com/Research/RR-2919.php.

Piotr Didyk, Tobias Ritschel, Elmar Eisemann, Karol Myszkowski, and Hans-Peter Seidel. A perceptual model for disparity. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 30(4):96:1–10, August 2011. ISSN 0730-0301. doi: 10.1145/2010324.1964991. URL http://www.mpi-inf.mpg.de/resources/DisparityModel/.

James Diebel and Sebastian Thrun. An application of Markov Random Fields to range sensing. In *Advances in Neural Information Processing Systems*, pages 291–298, December 2006.

Neil A. Dodgson. Variation and extrema of human interpupillary distance. In *Stereoscopic Displays and Virtual Reality Systems*, volume 5291 of *Proceedings of SPIE*, January 2004. doi: 10.1117/12.529999.

Neil A. Dodgson. Autostereoscopic 3D displays. *Computer*, 38(8):31–36, August 2005. ISSN 0018-9162. doi: 10.1109/MC.2005.252.

Jennifer Dolson, Jongmin Baek, Christian Plagemann, and Sebastian Thrun. Upsampling range data in dynamic environments. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1141–1148, June 2010. doi: 10.1109/CVPR.2010.5540086. URL http://graphics.stanford.edu/papers/upsampling_cvpr10/.

Frédo Durand and Julie Dorsey. Fast bilateral filtering for the display of high-dynamic-range images. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 21(3):257–266, July 2002. doi: 10.1145/566654.566574.

Geoffrey Egnal and Richard P. Wildes. Detecting binocular half-occlusions: Empirical comparisons of five approaches. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1127–1133, August 2002. ISSN 0162-8828. doi: 10.1109/TPAMI.2002.1023808.

Elmar Eisemann and Frédo Durand. Flash photography enhancement via intrinsic relighting. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 23(3):673–678, August 2004. doi: 10.1145/1186562.1015778.

Martin Eisemann, Bert De Decker, Marcus Magnor, Philippe Bekaert, Edilson de Aguiar, Naveed Ahmed, Christian Theobalt, and Anita Sellent. Floating textures. *Computer Graphics Forum (Proceedings of Eurographics)*, 27 (2):409–418, April 2008. URL http://graphics.tu-bs.de/projects/floating-textures/.

Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient belief propagation for early vision. *International Journal of Computer Vision*, 70(1):41–54, October 2006. ISSN 0920-5691. doi: 10.1007/s11263-006-7899-4.

Heather R. Filippini and Martin S. Banks. Limits of stereopsis explained by local cross-correlation. *Journal of Vision*, 9(1):8:1–18, January 2009. doi: 10.1167/9.1.8.

Jan Fischer, Dirk Bartz, and Wolfgang Straßer. Stylized augmented reality for improved immersion. In *Proceedings of the IEEE Conference on Virtual Reality (VR)*, pages 195–202, 325, March 2005. ISBN 0-7803-8929-8. doi: 10.1109/VR.2005.1492774. URL http://www.janfischer.com/pub_pages/pub-fischer05-vr.html.

Wilson S. Geisler and Karen D. Davila. Ideal discriminators in spatial vision: two-point stimuli. *Journal of the Optical Society of America A: Optics and Image Science*, 2(9):1483–1497, September 1985. doi: 10.1364/JOSAA.2.001483.

Margrit Gelautz, Efstathios Stavrakis, and Michael Bleyer. Stereo-based image and video analysis for multimedia applications. In *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, pages 998–1003, 2004.

James J. Gibson. The perception of visual surfaces. *The American Journal of Psychology*, 63(3):367–384, July 1950. doi: 10.2307/1418003.

Minglun Gong. Enforcing temporal consistency in real-time stereo estimation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 3953 of *Lecture Notes in Computer Science*, pages 564–577, May 2006. doi: 10.1007/11744078_44.

Minglun Gong and Yee-Hong Yang. Near real-time reliable stereo matching using programmable graphics hardware. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 924–931, June 2005. doi: 10.1109/CVPR.2005.246.

Amy A. Gooch, Bruce Gooch, Peter Shirley, and Elaine Cohen. A non-photorealistic lighting model for automatic technical illustration. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 447–452, July 1998. ISBN 0-89791-999-8. doi: 10.1145/280814.280950.

Amy A. Gooch, Jeremy Long, Li Ji, Anthony Estey, and Bruce S. Gooch. Viewing progress in non-photorealistic rendering through Heinlein's lens. In *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, pages 165–171, June 2010. ISBN 978-1-4503-0125-1. doi: 10.1145/1809939.1809959.

Bruce Gooch and Amy A. Gooch. *Non-Photorealistic Rendering*. A K Peters, 2001. ISBN 1568811330.

Stuart Green, David Salesin, Simon Schofield, Aaron Hertzmann, Peter Litwinowicz, Amy A. Gooch, Cassidy Curtis, and Bruce Gooch. Non-photorealistic rendering. In *SIGGRAPH Courses*, August 1999. URL http://mrl.nyu.edu/publications/npr-course1999/.

Mark Grundland, Chris Gibbs, and Neil A. Dodgson. Stylized multiresolution image representation. *Journal of Electronic Imaging*, 17(1):013009:1–17, April 2008. doi: 10.1117/1.2898894.

Paul Haeberli. Paint by numbers: abstract image representations. *Computer Graphics (Proceedings of SIGGRAPH)*, 24(4):207–214, August 1990. doi: 10.1145/97879.97902.

Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004. URL http://www.robots.ox.ac.uk/~vgg/hzbook/.

James Hays and Irfan A. Essa. Image and video based painterly animation. In *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, pages 113–120, June 2004. ISBN 1-58113-887-3. doi: 10.1145/987657.987676. URL http://www.cc.gatech.edu/cpl/projects/artstyling/.

Aaron Hertzmann. Painterly rendering with curved brush strokes of multiple sizes. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 453–460, July 1998. doi: 10.1145/280814.280951.

Aaron Hertzmann. A survey of stroke-based rendering. *IEEE Computer Graphics and Applications*, 23(4):70–81, July/August 2003. doi: 10.1109/MCG.2003.1210867.

Aaron Hertzmann and Ken Perlin. Painterly rendering for video and interaction. In *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, pages 7–12, June 2000. doi: 10.1145/340916.340917.

Aaron Hertzmann and Denis Zorin. Illustrating smooth surfaces. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 517–526, August 2000. doi: 10.1145/344779.345074.

Robert Herzog, Elmar Eisemann, Karol Myszkowski, and Hans-Peter Seidel. Spatio-temporal upsampling on the GPU. In *Proceedings of the Symposium on Interactive 3D Graphics and Games (I3D)*, pages 91–98, February 2010. doi: 10.1145/1730804.1730819.

David M. Hoffman, Ahna R. Girshick, Kurt Akeley, and Martin S. Banks. Vergence–accommodation conflicts hinder visual performance and cause visual fatigue. *Journal of Vision*, 8(3):1–30, March 2008. doi: 10.1167/8.3.33.

Berthold K. P. Horn. *Shape from Shading: A Method for Obtaining the Shape of a Smooth Opaque Object from One View*. PhD thesis, Massachusetts Institute of Technology, 1970. Published as technical report MIT-AITR-232.

Asmaa Hosni, Christoph Rhemann, Michael Bleyer, and Margrit Gelautz. Temporally consistent disparity and optical flow via efficient spatio-temporal filtering. In *Proceedings of the Pacific-Rim Symposium on Image and Video Technology (PSIVT)*, November 2011.

Ian P. Howard and Brian J. Rogers. *Seeing in Depth*. Oxford University Press, 2008. doi: 10.1093/acprof:oso/9780195367607.001.0001.

Peter A. Howarth. Potential hazards of viewing 3-D stereoscopic television, cinema and computer games: a review. *Ophthalmic and Physiological Optics*, 31(2):111–122, February 2011. ISSN 1475-1313. doi: 10.1111/j.1475-1313.2011.00822.x.

Michael Isard and John MacCormick. Dense motion and disparity estimation via loopy belief propagation. In *Proceedings of the Asian Conference on Computer Vision (ACCV)*, volume 3852 of *Lecture Notes in Computer Science*, pages 32–41, January 2006. doi: 10.1007/11612704_4.

Bela Julesz. Binocular depth perception without familiarity cues. *Science*, 145(3630):356–362, July 1964. doi: 10.1126/science.145.3630.356.

Robert D. Kalnins, Lee Markosian, Barbara J. Meier, Michael A. Kowalski, Joseph C. Lee, Philip L. Davidson, Matthew Webb, John F. Hughes, and Adam Finkelstein. WYSIWYG NPR: drawing strokes directly on 3D models. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 21(3):755–762, July 2002. ISSN 0730-0301. doi: 10.1145/566654.566648. URL http://gfx.cs.princeton.edu/pubs/Kalnins_2002_WND/.

Robert D. Kalnins, Philip L. Davidson, Lee Markosian, and Adam Finkelstein. Coherent stylized silhouettes. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 22(3):856–861, July 2003. doi: 10.1145/1201775.882355. URL http://gfx.cs.princeton.edu/pubs/Kalnins_2003_CSS/.

Henry Kang and Seungyong Lee. Shape-simplifying image abstraction. *Computer Graphics Forum (Proceedings of Pacific Graphics)*, 27(7):1773–1780, October 2008. doi: 10.1111/j.1467-8659.2008.01322.x.

Henry Kang, Seungyong Lee, and Charles K. Chui. Flow-based image abstraction. *IEEE Transactions on Visualization and Computer Graphics*, 15(1):62–76, January–February 2009. doi: 10.1109/TVCG.2008.81.

Atsushi Kasao and Kazunori Miyata. Algorithmic painter: a NPR method to generate various styles of painting. *The Visual Computer*, 22(1):14–27, January 2006. doi: 10.1007/s00371-005-0353-8.

Ramsin Khoshabeh, Stanley H. Chan, and Truong Q. Nguyen. Spatio-temporal consistency in video disparity estimation. In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2011. URL http://videoprocessing.ucsd.edu/~ramsin/research/disparity/.

Andreas Kolb, Erhardt Barth, Reinhard Koch, and Rasmus Larsen. Time-of-flight cameras in computer graphics. *Computer Graphics Forum*, 29(1):141–159, March 2010. doi: 10.1111/j.1467-8659.2009.01583.x.

Vladimir Kolmogorov and Ramin Zabih. Computing visual correspondence with occlusions using graph cuts. In *Proceedings of the International Conference on Computer Vision (ICCV)*, volume 2, pages 508–515, July 2001. doi: 10.1109/ICCV.2001.937668.

Frank L. Kooi and Alexander Toet. Visual comfort of binocular and 3D displays. *Displays*, 25(2–3):99–108, August 2004. doi: 10.1016/j.displa.2004.07.004.

Johannes Kopf, Michael F. Cohen, Dani Lischinski, and Matt Uyttendaele. Joint bilateral upsampling. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 26(3):96, July 2007. ISSN 0730-0301. doi: 10.1145/1276377.1276497.

Jan Eric Kyprianidis and Jürgen Döllner. Image abstraction by structure adaptive filtering. In *Proceedings of Theory and Practice of Computer Graphics (TPCG)*, pages 51–58, June 2008. URL http://www.kyprianidis.com/p/tpcg2008/.

Jan Eric Kyprianidis and Henry Kang. Image and video abstraction by coherence-enhancing filtering. *Computer Graphics Forum (Proceedings of Eurographics)*, 30(2), April 2011. URL http://www.kyprianidis.com/eg2011.html.

Jan Eric Kyprianidis, Henry Kang, and Jürgen Döllner. Image and video abstraction by anisotropic Kuwahara filtering. *Computer Graphics Forum (Proceedings of Pacific Graphics)*, 28(7):1955–1963, October 2009. doi: 10.1111/j.1467-8659.2009.01574.x. URL http://www.kyprianidis.com/p/pg2009/.

Marc T. M. Lambooij, Wijnand A. IJsselsteijn, Marten Fortuin, and Ingrid Heynderickx. Visual discomfort and visual fatigue of stereoscopic displays: A review. *Journal of Imaging Science and Technology*, 53(3):030201:1–14, May/June 2009. ISSN 10623701. doi: 10.2352/J.ImagingSci.Technol.2009.53.3.030201.

Douglas Lanman and Gabriel Taubin. Build your own 3D scanner: 3D photography for beginners. In *SIGGRAPH Courses*, pages 1–87, August 2009. doi: 10.1145/1667239.1667247.

Yunjin Lee, Lee Markosian, Seungyong Lee, and John F. Hughes. Line drawings via abstracted shading. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 26(3):18, July 2007. doi: 10.1145/1275808.1276400. URL http://cg.postech.ac.kr/research/line_drawings_via_abstracted_shading/.

Carlos Leung, Ben Appleton, Brian C. Lovell, and Changming Sun. An energy minimisation approach to stereotemporal dense reconstruction. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, volume 4, pages 72–75, August 2004. doi: 10.1109/ICPR.2004.1333708.

Chia-Kai Liang, Chao-Chung Cheng, Yen-Chieh Lai, Liang-Gee Chen, and Homer H. Chen. Hardware-efficient belief propagation. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 80–87, June 2009. doi: 10.1109/CVPR.2009.5206819.

Marvin Lindner, Andreas Kolb, and Klaus Hartmann. Data-fusion of PMD-based distance-information and high-resolution RGB-images. In *Proceedings of the International Symposium on Signals, Circuits and Systems (ISSCS)*, pages 121–124, July 2007. doi: 10.1109/ISSCS.2007.4292666.

Lenny Lipton. *Foundations of the Stereoscopic Cinema: A Study in Depth*. Van Nostrand Reinhold, 1982. ISBN 0442247249.

Peter Litwinowicz. Processing images and video for an impressionist effect. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 407–414, August 1997. ISBN 0-89791-896-7. doi: 10.1145/258734.258893.

Jingwan Lu, Pedro V. Sander, and Adam Finkelstein. Interactive painterly stylization of images, videos and 3D animations. In *Proceedings of the Symposium on Interactive 3D Graphics and Games (I3D)*, pages 127–134, February 2010. doi: 10.1145/1730804.1730825. URL http://gfx.cs.princeton.edu/pubs/Lu_2010_IPS/.

Thomas Luft and Oliver Deussen. Real-time watercolor illustrations of plants using a blurred depth test. In *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, pages 11–20, June 2006. doi: 10.1145/1124728.1124732. URL http://graphics.uni-konstanz.de/forschung/npr/watercolor/.

Tom Malzbender, Bennett Wilburn, Dan Gelb, and Bill Ambrisco. Surface enhancement using real-time photometric stereo and reflectance transformation. In *Proceedings of the Eurographics Symposium on Rendering*, pages 245–250, June 2006. doi: 10.2312/EGWR/EGSR06/245-250. URL http://graphics.stanford.edu/~wilburn/Papers/RealTimePhotometric.html.

Danijela Marković. *Image Stylization from Stereo Views of Natural Scenes*. PhD thesis, Vienna University of Technology, February 2007.

Danijela Marković and Margrit Gelautz. Drawing the real. In *Proceedings of the International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia (GRAPHITE)*, pages 237–243, December 2005. ISBN 1-59593-201-1. doi: 10.1145/1101389.1101436.

Danijela Marković and Margrit Gelautz. Comics-like motion depiction from stereo. In *Proceedings of the International in Central Europe on Computer Graphics, Visualisation and Computer Vision (WSCG)*, January 2006.

Danijela Marković, Efstathios Stavrakis, and Margrit Gelautz. Parameterized sketches from stereo images. In Amir Said and John G. Apostolopoulos, editors, *Image and Video Communications and Processing*, volume 5685 of *Proceedings of SPIE*, pages 783–791, March 2005. doi: 10.1117/12.585858.

George Mather. Image blur as a pictorial depth cue. *Proceedings of the Royal Society of London. Series B: Biological Sciences*, 263(1367):169–172, February 1996. doi: 10.1098/rspb.1996.0027.

Morgan McGuire, Henrik Halén, Jean-Francois St-Amour, Deano Calver, Aaron Thibault, Brian Martel, and Chandana Ekanayake. Stylized rendering in games. In *SIGGRAPH Courses*, July 2010. URL http://graphics.cs.williams.edu/courses/SRG10/.

Barbara J. Meier. Painterly rendering for animation. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 477–484, August 1996. ISBN 0-89791-746-4. doi: 10.1145/237170.237288.

Robert Neuman. Personal communication, 2008. Robert is Stereo Supervisor at Walt Disney Animation Studios.

Makoto Okabe, Gang Zeng, Yasuyuki Matsushita, Takeo Igarashi, Long Quan, and Heung-Yeung Shum. Single-view relighting with normal map painting. In *Proceedings of Pacific Graphics*, pages 27–34, October 2006. URL http://www.seman.cs.uec.ac.jp/~okabe/SingleViewRelighting/index.htm.

Alexandrina Orzan, Adrien Bousseau, Pascal Barla, and Joëlle Thollot. Structure-preserving manipulation of photographs. In *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, pages 103–110, August 2007. ISBN 978-1-59593-624-0. doi: 10.1145/1274871.1274888. URL http://maverick.inria.fr/Publications/2007/OBBT07/.

Stanley Osher and Leonid I. Rudin. Feature-oriented image enhancement using shock filters. *SIAM Journal on Numerical Analysis*, 27(4):919–940, 1990. doi: 10.1137/0727053.

Peter Ludvig Panum. *Physiologische Untersuchungen über das Sehen mit zwei Augen*. Schwersssche Buchhandlung, Kiel, 1858.

Sylvain Paris. Edge-preserving smoothing and mean-shift segmentation of video streams. In *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 5303 of *Lecture Notes in Computer Science*, pages 460–473, October 2008. doi: 10.1007/978-3-540-88688-4_34. URL http://people.csail.mit.edu/sparis/#eccv08.

Sylvain Paris and Frédo Durand. A fast approximation of the bilateral filter using a signal processing approach. *International Journal of Computer Vision*, 81:24–52, January 2009. doi: 10.1007/s11263-007-0110-8.

Sylvain Paris, Pierre Kornprobst, Jack Tumblin, and Frédo Durand. A gentle introduction to bilateral filtering and its applications. In *SIGGRAPH Classes*, August 2008. doi: 10.1145/1401132.1401134.

Sylvain Paris, Pierre Kornprobst, Jack Tumblin, and Frédo Duran. Bilateral filtering: Theory and applications. *Foundations and Trends in Computer Graphics and Vision*, 4(1):1–73, 2009. doi: 10.1561/0600000020.

Georg Petschnigg, Richard Szeliski, Maneesh Agrawala, Michael Cohen, Hugues Hoppe, and Kentaro Toyama. Digital photography with flash and no-flash image pairs. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 23(3):664–672, August 2004. doi: 10.1145/1186562.1015777.

Jonathan David Pfautz. *Depth perception in computer graphics*. PhD thesis, University of Cambridge Computer Laboratory, 2000. URL http://www.cl.cam.ac.uk/techreports/UCAM-CL-TR-546.html. Published as technical report UCAM-CL-TR-546.

Tuan Q. Pham and Lucas J. van Vliet. Separable bilateral filtering for fast video preprocessing. In *Proceedings of IEEE International Conference on Multimedia and Expo*, July 2005. doi: 10.1109/ICME.2005.1521458.

Carlos R. Ponce and Richard T. Born. Stereopsis. *Current Biology*, 18(18):R845–R850, September 2008. doi: 10.1016/j.cub.2008.07.006.

Emil Praun, Hugues Hoppe, Matthew Webb, and Adam Finkelstein. Real-time hatching. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 581–586, August 2001. doi: 10.1145/383259.383328.

Ravi Ramamoorthi and Pat Hanrahan. A signal-processing framework for inverse rendering. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 117–128, August 2001a. ISBN 1-58113-374-X. doi: 10.1145/383259.383271.

Ravi Ramamoorthi and Pat Hanrahan. An efficient representation for irradiance environment maps. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 497–500, August 2001b. ISBN 1-58113-374-X. doi: 10.1145/383259.383317. URL http://www.cs.berkeley.edu/~ravir/papers/envmap/.

Ramesh Raskar, Kar-Han Tan, Rogerio Feris, Jingyi Yu, and Matthew Turk. Non-photorealistic camera: depth edge detection and stylized rendering using multi-flash imaging. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 23(3):679–688, August 2004. doi: 10.1145/1186562.1015779.

Christoph Rhemann, Asmaa Hosni, Michael Bleyer, Carsten Rother, and Margrit Gelautz. Fast cost-volume filtering for visual correspondence and beyond. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2011. URL http://www.ims.tuwien.ac.at/research/costFilter/.

Whitman Richards. Stereopsis and stereoblindness. *Experimental Brain Research*, 10(4):380–388, August 1970. ISSN 0014-4819. doi: 10.1007/BF02324765.

Christian Richardt and Neil A. Dodgson. Voronoi video stylisation. In Brian Wyvill, editor, *Proceedings of Computer Graphics International (Short Papers)*, pages 103–108, May 2009. doi: 10.1145/1629739.1629752. URL http://richardt.name/voronoivideo/.

Christian Richardt, Jan Eric Kyprianidis, and Neil A. Dodgson. Stereo coherence in watercolour rendering. Poster at NPAR and Computational Aesthetics, June 2010a.

Christian Richardt, Douglas Orr, Ian Davies, Antonio Criminisi, and Neil A. Dodgson. Real-time spatiotemporal stereo matching using the dual-cross-bilateral grid. In Kostas Daniilidis, Petros Maragos, and Nikos Paragios, editors, *Proceedings of the European Conference on Computer Vision (ECCV)*, volume 6313 of *Lecture Notes in Computer Science*, pages 510–523, September 2010b. doi: 10.1007/978-3-642-15558-1_37. URL http://richardt.name/dcbgrid/.

Christian Richardt, Lech Świrski, Ian Davies, and Neil A. Dodgson. Predicting stereoscopic viewing comfort using a coherence-based computational model. In Douglas Cunningham and Tobias Isenberg, editors, *Proceedings of Computational Aesthetics*, pages 97–104, August 2011. doi: 10.1145/2030441.2030462. URL http://richardt.name/stereocomfort/.

Christian Richardt, Carsten Stoll, Neil A. Dodgson, Hans-Peter Seidel, and Christian Theobalt. Coherent spatio-temporal filtering, upsampling and rendering of RGBZ videos. *Computer Graphics Forum (Proceedings of Eurographics)*, 31(2), 2012. URL http://www.mpi-inf.mpg.de/resources/rgbz-camera/.

Brian Rogers and Maureen Graham. Motion parallax as an independent cue for depth perception. *Perception*, 8(2):125–134, 1979. doi: 10.1068/p080125.

Wilhelm Rollmann. Zwei neue stereoskopische Methoden. *Annalen der Physik*, 166:186–187, 1853. ISSN 1521-3889. doi: 10.1002/andp.18531660914.

Daniel Scharstein and Richard Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47(1–3):7–42, April 2002. ISSN 0920-5691. doi: 10.1023/A:1014573219977.

Daniel Scharstein and Richard Szeliski. High-accuracy stereo depth maps using structured light. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 195–202, June 2003. doi: 10.1109/CVPR.2003.1211354.

Johannes Schmid, Martin Sebastian Senn, Markus Gross, and Robert W. Sumner. OverCoat: An implicit canvas for 3D painting. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 30(4):28:1–10, August 2011. doi: 10.1145/2010324.1964923.

Pieter J. H. Seuntiëns, Lydia M. J. Meesters, and Wijnand A. IJsselsteijn. Perceptual attributes of crosstalk in 3D images. *Displays*, 26(4–5):177–183, October 2005. doi: 10.1016/j.displa.2005.06.005.

Mike Seymour and Robert Neuman. fxpodcast: Disney stereo tools. Podcast, February 2011. URL http://www.fxguide.com/fxpodcasts/robert-neuman-disney-stereo-tools/. Last accessed 2012-02-05.

Takashi Shibata, Joohwan Kim, David M. Hoffman, and Martin S. Banks. The zone of comfort: Predicting visual discomfort with stereo displays. *Journal of Vision*, 11(8):11, July 2011. doi: 10.1167/11.8.11.

Peter-Pike J. Sloan, William Martin, Amy A. Gooch, and Bruce Gooch. The lit sphere: a model for capturing NPR shading from art. In *Proceedings of Graphics Interface (GI)*, pages 143–150, June 2001. ISBN 0-9688808-0-0.

Stephen M. Smith and J. Michael Brady. SUSAN—a new approach to low level image processing. *International Journal of Computer Vision*, 23(1):45–78, May 1997. doi: 10.1023/A:1007963824710.

Noah Snavely, C. Lawrence Zitnick, Sing Bing Kang, and Michael Cohen. Stylizing 2.5-D video. In *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, pages 63–69, June 2006. doi: 10.1145/1124728.1124739.

Mario Costa Sousa, Brett Achorn, Daniel Teece, Sheelagh Carpendale, David S. Ebert, Bruce Gooch, Victoria Interrante, Lisa Streit, and Oleg Veryovka. Theory and practice of non-photorealistic graphics: Algorithms, methods, and production systems. In *SIGGRAPH Courses*, July 2003. URL http://www.siggraph.org/s2003/conference/courses/sousa.html.

Efstathios Stavrakis. *Stereoscopic Non-Photorealistic Rendering*. PhD thesis, Vienna University of Technology, December 2008.

Efstathios Stavrakis and Margrit Gelautz. Image-based stereoscopic painterly rendering. In *Proceedings of the Eurographics Symposium on Rendering*, June 2004.

Efstathios Stavrakis and Margrit Gelautz. Stereoscopic painting with varying levels of detail. In *Stereoscopic Displays and Virtual Reality Systems*, volume 5664 of *Proceedings of SPIE*, pages 450–459, January 2005a. doi: 10.1117/12.586702.

Efstathios Stavrakis and Margrit Gelautz. Computer generated stereoscopic artwork. In *Computational Aesthetics*, May 2005b.

Efstathios Stavrakis, Michael Bleyer, Danijela Marković, and Margrit Gelautz. Image-based stereoscopic stylization. In *Proceedings of the International Conference on Image Processing (ICIP)*, volume 3, pages 5–8, September 2005. doi: 10.1109/ICIP.2005.1530314.

Thomas Strothotte and Stefan Schlechtweg. *Non-Photorealistic Computer Graphics: Modeling, Rendering, and Animations*. Morgan Kaufmann, 2002. ISBN 1-55860-787-0. URL http://isgwww.cs.uni-magdeburg.de/pub/books/npr/.

Jian Sun, Nan-Ning Zheng, and Heung-Yeung Shum. Stereo matching using belief propagation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 25(7):787–800, July 2003. ISSN 0162-8828. doi: 10.1109/TPAMI.2003.1206509.

Technicolor. Certifi3D 15-point quality checklist chart. Poster at Consumer Electronics Show, January 2011. URL http://www.itbroadcastanddigitalcinema.com/docs/2011-01-08_Technicolor_Certifi3D.pdf. Last accessed 2012-02-05.

Dejan Todorović. Gestalt principles. *Scholarpedia*, 3(12):5345, 2008. doi: 10.4249/scholarpedia.5345.

Corey Toler-Franklin, Adam Finkelstein, and Szymon Rusinkiewicz. Illustration of complex real-world objects using images with normals. In *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, pages 111–119, August 2007. ISBN 978-1-59593-624-0. doi: 10.1145/1274871.1274889.

Carlo Tomasi and R. Manduchi. Bilateral filtering for gray and color images. In *Proceedings of the International Conference on Computer Vision (ICCV)*, pages 839–846, January 1998. ISBN 81-7319-221-9. doi: 10.1109/ICCV.1998.710815.

Christopher W. Tyler. Binocular vision. In William Tasman and Edward A Jaeger, editors, *Duane's Foundations of Clinical Ophthalmology*, volume 2. Lippincott Williams & Wilkins, 2004.

Romain Vergne, David Vanderhaeghe, Jiazhou Chen, Pascal Barla, Xavier Granier, and Christophe Schlick. Implicit brushes for stylized line-based rendering. *Computer Graphics Forum (Proceedings of Eurographics)*, 30(2): 513–522, April 2011. doi: 10.1111/j.1467-8659.2011.01892.x.

Vibhav Vineet and P.J. Narayanan. CUDA cuts: Fast graph cuts on the GPU. In *Proceedings of CVPR Workshops*, June 2008. doi: 10.1109/CVPRW.2008.4563095.

Björn N. S. Vlaskamp, Heather R. Filippini, and Martin S. Banks. Image-size differences worsen stereopsis independent of eye position. *Journal of Vision*, 9(2):17:1–13, February 2009. doi: 10.1167/9.2.17.

Jue Wang, Yingqing Xu, Heung-Yeung Shum, and Michael F. Cohen. Video tooning. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 23(3):574–583, August 2004a. ISSN 0730-0301. doi: 10.1145/1015706.1015763.

Liang Wang, Miao Liao, Minglun Gong, Ruigang Yang, and D. Nister. High-quality real-time stereo using adaptive cost aggregation and dynamic programming. In *Proceedings of the International Symposium on 3D Data Processing, Visualization, and Transmission (3DPVT)*, pages 798–805, June 2006. doi: 10.1109/3DPVT.2006.75.

Oliver Wang, Martin Fuchs, Christian Fuchs, James Davis, Hans-Peter Seidel, and Hendrik P. A. Lensch. A context-aware light source. In *Proceedings of the International Conference on Computational Photography (ICCP)*, March 2010. doi: 10.1109/ICCPHOT.2010.5585091.

Zhou Wang, Alan C. Bovik, Hamid R. Sheikh, and Eero P. Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE Transactions on Image Processing*, 13(4):600–612, April 2004b. doi: 10.1109/TIP.2003.819861.

Ben Weiss. Fast median and bilateral filtering. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 25(3): 519–526, July 2006. ISSN 0730-0301. doi: 10.1145/1141911.1141918. URL http://www.shellandslate.com/fastmedian.html.

Fang Wen, Qing Luan, Lin Liang, Ying-Qing Xu, and Heung-Yeung Shum. Color sketch generation. In *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, pages 47–54, June 2006. doi: 10.1145/1124728.1124737.

Andreas Wenger, Andrew Gardner, Chris Tchou, Jonas Unger, Tim Hawkins, and Paul Debevec. Performance relighting and reflectance transformation with time-multiplexed illumination. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 24(3):756–764, July 2005. ISSN 0730-0301. doi: 10.1145/1073204.1073258.

Max Wertheimer. Untersuchungen zur Lehre von der Gestalt. II. *Psychological Research*, 4:301–350, January 1923. ISSN 0340-0727. doi: 10.1007/BF00410640.

Charles Wheatstone. Contributions to the physiology of vision.—Part the first. On some remarkable, and hitherto unobserved, phenomena of binocular vision. *Philosophical Transactions of the Royal Society of London*, 128:371–394, 1838. URL http://www.jstor.org/stable/108203.

Oliver Williams, Michael Isard, and John MacCormick. Estimating disparity and occlusions in stereo video sequences. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 250–257, June 2005. doi: 10.1109/CVPR.2005.146.

Georges Winkenbach and David H. Salesin. Computer-generated pen-and-ink illustration. In *Proceedings of the Annual Conference on Computer Graphics and Interactive Techniques (SIGGRAPH)*, pages 91–100, July 1994. ISBN 0-89791-667-0. doi: 10.1145/192161.192184.

Holger Winnemöller. XDoG: Advanced image stylization with eXtended Difference-of-Gaussians. In *Proceedings of the International Symposium on Non-Photorealistic Animation and Rendering (NPAR)*, August 2011. doi: 10.1145/2024676.2024700.

Holger Winnemöller, Sven C. Olsen, and Bruce Gooch. Real-time video abstraction. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 25(3):1221–1226, July 2006. doi: 10.1145/1179352.1142018.

Robert J. Woodham. Photometric method for determining surface orientation from multiple images. *Optical Engineering*, 19(1):139–144, January/February 1980.

Qingxiong Yang, Ruigang Yang, James Davis, and David Nistér. Spatial-depth super resolution for range images. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2007. doi: 10.1109/CVPR.2007.383211.

Qingxiong Yang, Chris Engels, and Amir Akbarzadeh. Near real-time stereo for weakly-textured scenes. In *Proceedings of the British Machine Vision Conference (BMVC)*, September 2008.

Qingxiong Yang, Kar-Han Tan, and Narendra Ahuja. Real-time O(1) bilateral filtering. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009. doi: 10.1109/CVPR.2009.5206542.

Kuk-Jin Yoon and In So Kweon. Adaptive support-weight approach for correspondence search. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(4):650–656, April 2006. ISSN 0162-8828. doi: 10.1109/TPAMI.2006.70.

Li Zhang, Brian Curless, and Steven M. Seitz. Spacetime stereo: shape recovery for dynamic scenes. In *Proceedings of the International Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 2, pages 367–374, June 2003. doi: 10.1109/CVPR.2003.1211492. URL http://grail.cs.washington.edu/projects/ststereo/.

Li Zhang, Noah Snavely, Brian Curless, and Steven M. Seitz. Spacetime faces: high resolution capture for modeling and animation. *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 23(3):548–558, August 2004. ISSN 0730-0301. doi: 10.1145/1015706.1015759.

Ruo Zhang, Ping-Sing Tsai, James Edwin Cryer, and Mubarak Shah. Shape-from-shading: a survey. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(8):690–706, August 1999. ISSN 0162-8828. doi: 10.1109/34.784284.

Jin Zhu. Temporally consistent disparity estimation using PCA dual-cross-bilateral grid. Master's thesis, Technische Universiteit Eindhoven, 2011. URL http://repository.tue.nl/721239.